

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR
ZHEN ZHAO

ÉTUDE DES PROTOCOLES DE COMMUNICATION POUR LES SYSTÈMES DE
GESTION DANS LE CONTEXTE DES RÉSEAUX INTELLIGENTS

MARS 2017

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Résumé

En raison d'une demande croissante d'électricité et des problèmes environnementaux qu'elle cause, il est nécessaire d'exploiter l'énergie de manière efficace et effective. Un système de gestion d'énergie résidentielle HEMS (acronyme de Home Energy Management System) représente une solution permettant d'intégrer le concept à la vie quotidienne. HEMS est un outil qui contribue à déplacer le pic de demande d'électricité et de réduire la consommation d'énergie en fonction de l'information fournie par l'utilité électrique. Grâce aux produits intelligents variés qui se trouvent dans un HEMS, les consommateurs peuvent mieux comprendre et gérer leur utilisation énergétique. Normalement, les produits qui se trouvent dans un HEMS incluent : 1) les appareils de contrôle de charge permettant au consommateur de contrôler à distance ses électroménagers (par exemple : thermostats, chauffe-eau, éclairage) à distance; 2) les afficheurs (ou moniteurs) domestiques qui offrent des interfaces fournissant les informations sur des appareils dans le réseau résidentiel, qui permettent, dans certains cas, de transmettre les commandes pour contrôler les appareils.

L'un des défis à implémenter HEMS est la connexion dans le réseau résidentiel avec les appareils intelligents commerciaux pour établir un HEMS conçu par différents fabricants avec diverses technologies de communication. À part le Wi-Fi qui est le plus populaire, il existe aussi des protocoles de communication émergents, tels que HomePlug, Bluetooth, Z-Wave, ZigBee, etc. Il faut donc posséder la connaissance des caractéristiques des protocoles et les scénarios qu'ils visent. Un réseau résidentiel comporte probablement des

produits qui fonctionnent selon différents protocoles. Comment les appareils peuvent-ils coexister dans le même réseau et être contrôlés par une interface? De plus, il faut tenir compte du fait que la plupart des produits sont fermés, c'est-à-dire que les développeurs ne peuvent pas implémenter des applications fondées sur leurs besoins.

L'adoption de HEMS devient nécessaire pour la gestion énergétique résidentielle et pour le maintien du confort des clients. La passerelle constitue un élément essentiel dans un HEMS, elle représente le nœud du réseau public et du réseau résidentiel. Toutes les informations concernant les appareils dans un HEMS doivent être réunies dans une passerelle, conséquemment la passerelle du futur doit posséder la capacité d'intégrer différents protocoles et technologies.

Le présent travail consiste principalement à étudier la connectivité dans les systèmes de gestion locale résidentielle (HEMS); plus spécifiquement : établir la comparaison entre les technologies de communication (HomePlug, Wi-Fi, Bluetooth, Z-Wave et ZigBee); étudier le protocole d'application : SEP (Smart Energy Profile); ainsi que les expérimentations sur une architecture élémentaire pour la gestion locale basée sur la technologie ZigBee.

Nous avons réalisé dans le cadre de ce projet des études sur les caractéristiques des technologies de communication, ce qui a permis d'identifier ZigBee comme étant parmi l'une des technologies les plus appropriées pour l'implantation de HEMS. Conséquemment, trois expérimentations ont été effectuées afin de tester la performance de ZigBee, de transformer un module de ZigBee à SEP, de développer une application Web pour la surveillance et le contrôle de charges.

Remerciements

En préambule de ce mémoire, je souhaite remercier mon directeur de recherche Kodjo Agbossou, directeur de l'École d'ingénierie de l'Université du Québec à Trois-Rivières (UQTR), de m'avoir donné la chance pour participer à ce projet, de son soutien et sa patience pendant la période de mes études de maîtrise.

Je voudrais remercier également Dr Alben Cardenas pour l'orientation, ses conseils de mon travail, et la révision de mon mémoire. Grâce à son aide que j'ai pu accomplir mon travail. Je tiens aussi à remercier toute l'équipe de recherche pour avoir partagé les idées de recherche et répondre à mes questions.

Table des matières

| | |
|---|-----|
| Résumé..... | ii |
| Remerciements..... | iv |
| Table des matières..... | v |
| Liste des tableaux..... | ix |
| Liste des figures..... | x |
| Liste des symboles..... | xii |
| Chapitre 1 - Introduction..... | 1 |
| 1.1 Contexte et objectifs..... | 2 |
| 1.2 Méthodologie..... | 3 |
| 1.3 Structure du mémoire..... | 5 |
| Chapitre 2 - HEMS (Home Energy Management System)..... | 6 |
| 2.1 L'architecture d'un HEMS..... | 7 |
| 2.2 Les avantages d'un HEMS..... | 8 |
| 2.3 L'importance de passerelle pour HEMS..... | 9 |
| 2.4 Conclusion..... | 12 |
| Chapitre 3 - Comparaison de HomePlug, Wi-Fi, Bluetooth, Z-Wave, ZigBee..... | 13 |

| | | |
|-------|--|----|
| 3.1 | Communication par courants porteurs en ligne : HomePlug..... | 14 |
| 3.1.1 | HomePlug Green PHY..... | 14 |
| 3.1.2 | Produits commerciaux disponibles dans la technologie HomePlug | 15 |
| 3.2 | Wi-Fi | 17 |
| 3.2.1 | Wi-Fi CERTIFIED n..... | 18 |
| 3.2.2 | Produits commerciaux disponibles dans la technologie Wi-Fi..... | 19 |
| 3.3 | Bluetooth | 20 |
| 3.3.1 | Bluetooth 4.0 + (ou Bluetooth Smart; Bluetooth Low Energy)..... | 22 |
| 3.3.2 | Produits commerciaux disponibles dans la technologie Bluetooth..... | 22 |
| 3.4 | Z-Wave | 24 |
| 3.4.1 | Les caractéristiques de Z-Wave | 24 |
| 3.4.2 | Produits commerciaux disponibles dans la technologie Z-Wave | 26 |
| 3.5 | ZigBee | 28 |
| 3.5.1 | Caractéristiques du protocole ZigBee | 29 |
| 3.5.2 | Produits commerciaux disponibles dans la technologie ZigBee..... | 31 |
| 3.6 | Comparaison des protocoles de communication pour le système de gestion locale | 33 |
| 3.7 | Conclusion..... | 37 |

| | |
|---|----|
| Chapitre 4 - Smart Energy Profile 2.0 (SEP2.0) | 38 |
| 4.1 Introduction au SEP2.0..... | 38 |
| 4.2 Architecture de SEP2.0 et les technologies supportées dans la couche d'application | 40 |
| 4.3 La sécurité | 41 |
| 4.4 Ressource..... | 42 |
| 4.5 Les fonctions principales | 43 |
| 4.6 Conclusion..... | 48 |
| Chapitre 5 - L'application d'interopérabilité de la gestion d'énergie résidentielle..... | 50 |
| 5.1 Matériel utilisé..... | 51 |
| 5.2 Construction d'un réseau ZigBee pour contrôler la température d'une chambre | 53 |
| 5.2.1 L'environnement de programmation -- ESP..... | 54 |
| 5.2.2 L'interface web locale pour la configuration des dispositifs - XBee | 55 |
| 5.2.3 La programmation de l'application..... | 58 |
| 5.3 Construction d'un réseau ZigBee-SE | 63 |
| 5.4 Développement d'une application pour la gestion locale avec ZigBee | 64 |
| 5.4.1 Structure du réseau pour l'application | 65 |

| | | |
|---|--|----|
| 5.4.2 | Processus de développement de l'application Web | 66 |
| 5.4.3 | Le résultat de l'expérimentation du développement d'une application web | 73 |
| 5.5 | Conclusion..... | 75 |
| Chapitre 6 - Conclusion | | 77 |
| Bibliographie..... | | 80 |
| Annexe A – Manuel d'utilisation..... | | 86 |
| Annexe B – Code Source des Expérimentations | | 97 |

Liste des tableaux

| | |
|--|----|
| Tableau 3-1 Les caractéristiques d’HomePlug Green PHY[8] | 15 |
| Tableau 3-2 Les spécifications de l’adaptateur TL – PA4020P [23] | 16 |
| Tableau 3-3 Les spécifications du routeur NB318S[24] | 17 |
| Tableau 3-4 Les spécifications de standard IEEE 802.11a/b/g/n [25] | 18 |
| Tableau 3-5 Mydlink - Wi-Fi Motion sensor (DCH-S150) | 19 |
| Tableau 3-6 Lyric Round Wi-Fi Thermostat | 20 |
| Tableau 3-7 La comparaison de Bluetooth classique et BLE [33][34] | 22 |
| Tableau 3-8 Spécifications de Robosmart ampoule LED [35] | 23 |
| Tableau 3-9 Spécifiques d’EVE Energy switch&meter | 24 |
| Tableau 3-10 Spécifications de Vision Pro Z-Wave thermostat | 27 |
| Tableau 3-11 2-boutons contrôleur avec interrupteur | 28 |
| Tableau 3-12 Les caractéristiques de ZigBee[32] | 30 |
| Tableau 3-13 Spécification du capteur MCT-340 SMA | 32 |
| Tableau 3-14 Spécification de Wireless Area Controller (WAC50) | 32 |
| Tableau 3-15 Comparaison des technologies de communication | 33 |

Liste des figures

| | |
|--|----|
| Figure 2.1 L'architecture d'un HEMS[8][9] | 8 |
| Figure 3.1 Topologie de piconet | 21 |
| Figure 3.2 Topologie de scatternet..... | 21 |
| Figure 3.3 le protocole de Z-Wave[38]..... | 25 |
| Figure 3.4 Topologie de Z-Wave..... | 26 |
| Figure 3.5 La structure du protocole ZigBee[42]..... | 29 |
| Figure 4.1 Les structures des protocoles ZigBee, module OSI, SEP1.x et SEP2.0 | 39 |
| Figure 4.2 Un exemple de la fonctionnalité de mesurage[51] | 45 |
| Figure 4.3 Un exemple de la fonctionnalité de tarification[51] | 46 |
| Figure 4.4 Un exemple de la fonctionnalité de DRLC[51] | 48 |
| Figure 5.1 Les produits de Digi qui utilisés dans les expérimentations..... | 52 |
| Figure 5.2 La structure d'un réseau de ZigBee pour le contrôle de température..... | 53 |
| Figure 5.3 La structure de Digi ESP | 54 |
| Figure 5.4 L'interface de la gestion de dispositif..... | 55 |
| Figure 5.5 L'interface de web local pour la gestion de dispositif..... | 56 |
| Figure 5.6 La page de la configuration de XBee | 57 |
| Figure 5.7 Configurer le pin d'entrée/sortie (I/O) par l'interface Web..... | 58 |
| Figure 5.8 ESP pour programmation de Python | 60 |
| Figure 5.9 Liste des fichiers enregistrés dans la passerelle..... | 61 |
| Figure 5.10 Activer le programme | 61 |

| | |
|---|----|
| Figure 5.11 Les résultats d'expérimentation a) l'état de la prise intelligente (1:ON, 0:OFF) et b) la température de chambre | 62 |
| Figure 5.12 Réseau de SEP | 64 |
| Figure 5.13 La structure d'application web de HEMS | 65 |
| Figure 5.14 L'organigramme de programmation..... | 67 |
| Figure 5.15 La structure simple de HTML | 68 |
| Figure 5.16 La relation de MTV | 70 |
| Figure 5.17 Un exemple de WSGI application | 72 |
| Figure 5.18 Un exemple de WSGI serveur | 72 |
| Figure 5.19 L'interface web d'application interopérable dans HEMS | 74 |

Liste des symboles

| | |
|------|---|
| AC | Air conditionné |
| AMI | <i>Advanced Metering Standard Infrastructure</i> |
| APL | Couche d'application (<i>Application Layer</i>) |
| AES | Standard de chiffrement avancé (<i>Advanced Encryption Standard</i>) |
| API | Interface de programmation applicative (<i>Application Programming Interface</i>) |
| ADC | Convertisseur analogique-numérique (<i>Analog to Digital Converter</i>) |
| BLE | <i>Bluetooth Low Energy</i> |
| CPL | Courants Porteurs en Ligne (<i>PLC : Power Line Carrier</i>) |
| CIM | Modèle de données unifié (<i>Common Information Model</i>) |
| CSS | Les feuilles de style en cascade (<i>Cascading Style Sheets</i>) |
| CGI | Interface de passerelle commune (<i>Common Gateway Interface</i>) |
| DR | Réponse à la demande (<i>Demand Response</i>) |
| DSSS | Étalement de spectre à séquence directe (<i>Direct Sequence Spread Spectrum</i>) |
| DRLC | Gestion de la Demande et Contrôle Direct de Charge (<i>Demand Reponse Load Control</i>) |
| EV | Véhicule électrique (<i>Electric Vehicle</i>) |

| | |
|------|---|
| ESI | <i>Energy Services Interface</i> |
| FHSS | Étalement de spectre par saut de fréquence (<i>Frequency-Hopping Spread Spectrum</i>) |
| GTL | Grille tarifaire réglementée |
| GUI | Interface graphique (<i>Graphical User Interface</i>) |
| HAN | Réseau résidentiel (<i>Home Area Network</i>) |
| HEMS | <i>Home Energy Management System</i> |
| HTTP | Protocole de transfert hypertexte (<i>HyperText Transfer Protocol</i>) |
| HTML | <i>HyperText Markup Language</i> |
| IoT | Internet des objets (<i>Internet of Thing</i>) |
| IDE | <i>Integrated Development Environment</i> |
| MDE | Maîtrise de la demande en énergie (<i>Energy Demand Management</i>) |
| MIMO | Entrées multiples, sorties multiples (<i>Multiple-Input Multiple-Output</i>) |
| MAC | Contrôle d'accès au support (<i>Media access control</i>) |
| MVC | Modèle-vue-contrôleur (<i>Model-view-controller</i>) |
| MTV | Modèle-Template-Vue |
| NWK | Network |
| PAN | Réseau personnel sans fil (<i>Personnel Area Network</i>) |
| PHY | Couche physique (<i>Physical layer</i>) |

| | |
|------|---|
| REST | <i>Representational state transfer</i> |
| SEP2 | <i>Smart Energy Profile 2</i> |
| TOU | <i>Time-of-Use rates</i> |
| TLS | <i>Transport Layer Security</i> |
| URI | <i>Uniform Resource Identifier</i> |
| UML | <i>Unified Modeling Language</i> |
| WSDL | <i>Web Application Description Language</i> |
| WSGI | <i>Web Server Gateway Interface</i> |
| XSD | <i>XML Schema Definition</i> |

Chapitre 1 - Introduction

L'énergie est un élément essentiel du développement de la société, par conséquent, l'exigence envers l'énergie augmente continuellement jusqu'à présent. Pourtant, à cause du développement excessif, le déficit énergétique est devenu un problème considérable et mondial. Aussi, le réchauffement climatique, l'accroissement des émissions de gaz à effet de serre et les autres problèmes environnementaux conduisent les gouvernements et utilities électriques à déployer des efforts vers l'exploitation des énergies renouvelables, ainsi que l'amélioration du système de transport et de distribution d'énergie, pour accroître leurs efficacités et les flexibilités [1]. Toutes les demandes réunies ont stimulé l'émergence du concept des Réseaux Intelligents (plus connu sous le nom de Smart Grid). Le réseau intelligent fonctionne avec les technologies numériques et d'information et fournit un moyen plus moderne de déployer les techniques de la consommation d'énergie. Il tend à remplacer le réseau traditionnel [2].

Le réseau intelligent, par rapport au réseau traditionnel, assure un flux bidirectionnel d'électricité et d'information qui permet aux consommateurs d'accéder aux produits d'électricité et de communiquer avec les services d'utilité publique. Par exemple, un réseau qui emploie la notion de Gestion de la Demande Locale (DSM, en anglais Demand Side Management) peut réduire le pic de la demande d'énergie jusqu'à 17% [3]. D'un côté, les électroménagers qui ont la capacité de communication construisent un HAN (pour Home Area Network), d'autre côté, il faut que l'utilité et le HAN communiquent ensemble et

s'envoient des messages [4]. Par conséquent, les développeurs sont confrontés à quelques défis quant à la communication: 1) le choix de la meilleure technologie de communication pour les appareils dans la maison; 2) la coordination et le contrôle des appareils intelligents; 3) la communication bidirectionnelle complexe entre les HAN et les utilités. Ces trois points seront abordés dans le contenu de ce mémoire.

1.1 Contexte et objectifs

L'électricité est la forme d'énergie la plus largement utilisée dans notre vie quotidienne, le potentiel de réduire la consommation de l'électricité dans la maison est considérable. Donc, qu'est-ce qu'on peut faire pour économiser l'électricité et, en même temps, ne rien sacrifier de notre confort? Une des solutions serait de construire un HEMS (Home Energy Management System) qui peut gérer efficacement la consommation des électroménagers dans un réseau domestique (HAN) de communication intelligent, et cela semble être la tendance d'avenir.

Le HEMS est un système qui fonctionne avec les appareils de DR (Demand Response), profite des progrès de la communication, de l'informatique, et des technologies d'affichage. Il permet d'effectuer le contrôle automatisé (ou semi-automatisé) des charges basé sur les signaux de l'utilité, la priorité de la charge, ainsi que la préférence des consommateurs. De plus, il assiste les clients afin qu'ils puissent comprendre et gérer leur consommation d'énergie. En général, un HEMS inclut : (i) une unité HEMS, qui fournit les fonctions de surveillance et de contrôle pour le propriétaire; (ii) des contrôleurs de charges qui permettent de gérer (ou de mesurer seulement) la consommation électrique de certains appareils, comme le système de chauffage, climatisation et ventilation (HVAC acronyme

de Heating Ventilation and Air-Conditioning), le chauffe-eau, les véhicules électriques (EV), etc.; et (iii) une passerelle résidentielle.

Pour installer un HEMS dans un HAN, il faut que ces trois éléments communiquent entre eux. Un certain nombre de technologies de communication sont accessibles et, dans ce cas, ce mémoire a pour objectif principal l'étude et la comparaison des différents protocoles et standards de communication, existants et émergents, pouvant être utilisés pour la gestion locale de la demande. Notamment, le protocole d'application : Smart Energy Profile 2 (SEP2) est étudié, les performances et l'adaptabilité de SEP2 pour le contrôle et la gestion dans HEMS sont analysées; de plus, les aspects qui doivent être améliorés afin de remplir les besoins pour la gestion domestique sont proposés. Comme la passerelle est un élément indispensable dans un réseau intelligent, un autre objectif de ce mémoire consiste à exposer l'importance et les fonctions principales de la passerelle.

Enfin, le dernier objectif, mais non le moindre, c'est de choisir une technologie de communication (ZigBee) et bâtir un réseau permettant le contrôle et la surveillance de charges résidentielles. Des études théoriques, des modélisations, des simulations et des validations expérimentales sont menées afin d'établir des points de l'analyse et afin de réaliser les fonctions pour l'implantation des systèmes intelligents de gestion de la demande.

1.2 Méthodologie

Pour réaliser les travaux, la méthodologie suivante a été adoptée :

1. Analyse préliminaire des technologies existantes pour la communication locale dans le contexte des réseaux électriques intelligents.

- a. Comparaison des technologies sans fil (ZigBee, Z-Wave, Wi-Fi) et à courant porteur (PLC).
 - b. Étude des passerelles (Gateways) « disponibles ou proposées dans la littérature » pour la gestion locale (pour HEMS).
 - i. Technologies intégrées dans les passerelles.
 - ii. Fonctions supportées.
 - iii. Exemples d'application dans l'industrie et dans la littérature.
 - iv. Identification des améliorations possibles quant aux fonctions permettant de mieux répondre aux besoins de gestion locale (désagrégation, contrôle, gestion de charges).
 - c. Étude du protocole d'application pour la gestion locale « Smart Energy ».
 - i. Caractéristiques et définitions selon les versions disponibles.
 - ii. Identification des « limites / contraintes » principalement en lien avec le taux de transmission, le type d'information échangée, et les applications qui ne sont pas supportées.
 - iii. Exemples d'application dans l'industrie et dans la littérature.
 - iv. Possibles améliorations.
2. Analyse des performances du réseau ZigBee pour le système de gestion locale d'énergie (HEMS).
- a. Proposition et mise en œuvre d'un système de tests expérimental permettant l'évaluation de ZigBee dans un scénario de gestion locale.

- b. Validation expérimentale sur les ZigBee produits commerciaux, réaliser les fonctions pour le système intelligents de gestion de la demande, et analyser les données par Matlab.
- c. Développements d'une application web pour surveiller et contrôler les ZigBee produits commerciaux qui composent le réseau local.

1.3 Structure du mémoire

Le mémoire se compose de six chapitres, le chapitre 2 décrit le système de gestion d'énergie dans la maison (Home Energy Management System : HEMS) en général, ainsi que l'importance des passerelles (Gateways) et la structure d'un réseau domestique de communication (HAN). Le chapitre 3 présente les caractéristiques des technologies de communication qui conviennent à l'environnement de HAN. Les technologies mentionnées incluent: HomePlug, Ethernet, Wi-Fi, Z-Wave et ZigBee. Cette partie parle aussi de la comparaison des différentes technologies et propose le meilleur choix pour la communication au niveau résidentiel. Dans le chapitre 4, le concept de « Smart Energy Profile » est présenté. Le chapitre 5 présente les expérimentations qui ont permis d'analyser les performances des produits ZigBee et le processus du développement d'une application pour le contrôle et la surveillance dans un système résidentiel. Finalement, le chapitre 6 est consacré aux conclusions et recommandations du mémoire.

Chapitre 2 - HEMS (Home Energy Management System)

Les problèmes d'énergie ont soulevé une grande attention depuis des années, ils causent plusieurs difficultés environnementales, comme la pollution de l'air, l'émission de gaz à effet de serre, la consommation de source, etc. En fait, une grande quantité d'énergie est consommée dans les résidences, particulièrement par les appareils électroménagers, par les systèmes de chauffage et par l'éclairage. Selon une enquête menée par un département de Statistique Canada, en 2011, les ménages canadiens ont consommé 1.4 million Térajoules (TJ) d'énergie [5]. Il est évident que la consommation d'énergie résidentielle a une influence significative sur l'environnement. Généralement, le réseau intelligent (Smart Grid) représente la tendance comme moyen efficient, pratique, et économique de consommer l'énergie. Le réseau domestique de communication HAN (Home Area Network), en tant que partie essentielle d'un réseau intelligent, est construit par les appareils intelligents résidentiels et offre une façon de surveiller et contrôler les appareils dans l'optique de gérer l'utilisation de l'énergie. En plus, un HAN intelligent doit comporter des systèmes de gestion de l'énergie, plus connue sous le nom de HEMS (Home Energy Management System), afin de transformer l'habitude de la consommation d'énergie des consommateurs par l'offre d'un système de gestion visible, qui retourne l'information des dispositifs en temps-réel, ce qui aide à réduire la quantité d'énergie dépensée dans une résidence.

Ce chapitre présente l'architecture d'un HEMS et ses avantages, ainsi que l'importance de la passerelle, qui est nécessaire pour la connexion des différents éléments du système de gestion.

2.1 L'architecture d'un HEMS

Un HEMS est formé par plusieurs parties. Normalement, les cinq composants suivants sont présents dans un HEMS [6]. Une architecture typique est présentée à la figure 2.1.

- **Capteurs** : un HEMS peut intégrer plusieurs capteurs qui possèdent différentes fonctions, par exemple les capteurs pour détecter la température, le courant, la lumière, ou l'humidité... Les capteurs envoient les données détectées à un système central, les appareils intelligents peuvent alors être gérés en se basant sur ces données.
- **Appareils de mesure** : le plus souvent, dans une maison, il existe des compteurs d'électricité, d'eau, de gaz. Le système de mesurage avancé, connu sous le nom de AMI (Advanced Metering Infrastructure), qui contribue à recueillir les informations de consommation, analyse les données, donne la communication entre l'utilité et la maison, et la facture en temps-réel.[7]
- **Appareils intelligents** : un appareil intelligent se distingue des simples appareils par ses capacités de communication et ses caractéristiques programmables; il peut être surveillé et contrôlé à distance. Il est plus efficace, et commode pour aider le client à prendre de bonnes habitudes dans l'utilisation de l'énergie, et à économiser l'énergie.

- Technologie de communication : les technologies de communication, soit avec fil, soit sans fil, ce sont les conditions nécessaires pour créer un réseau local, ainsi que pour communiquer entre les capteurs, les mesures, les appareils intelligents, et les autres dispositifs qui peuvent communiquer.
- Le système de la gestion d'énergie : ce système fournit une interface qui est une plate-forme pour les clients afin de surveiller le HEMS, d'envoyer les commandes à une destination désignée. Un logiciel est incorporé pour gérer le HEMS par le programme spécifique.

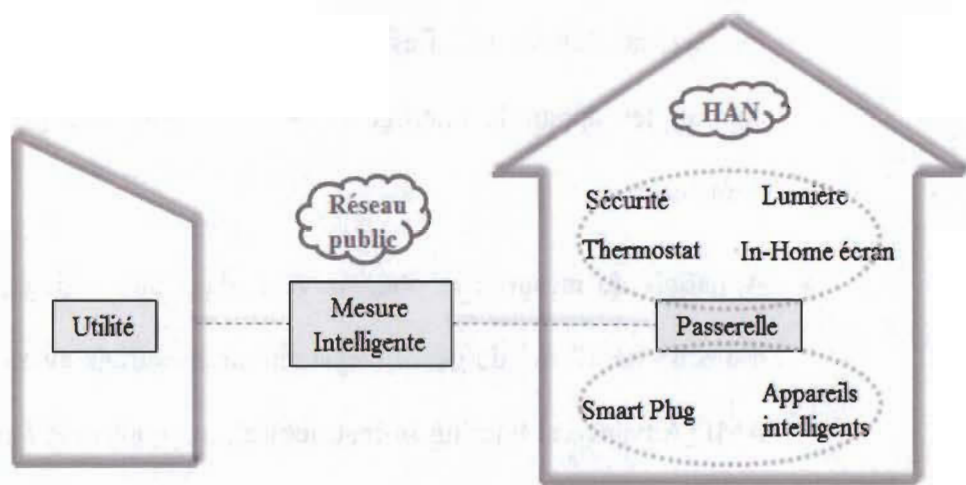


Figure 2.1 Architecture typique d'un HEMS[8], [9]

2.2 Les avantages d'un HEMS

Les avantages d'un HEMS concernent deux aspects: à court terme et à long terme.

Pour le court terme, les consommateurs sont les bénéficiaires qui profitent du HEMS directement. Tout d'abord, un HEMS permet aux consommateurs de réduire les coûts d'énergie. Selon la recherche de [10], la facture en temps-réel aide les gens à diminuer leur

dépense de consommation de 5 % à 15 %. D'ailleurs, un moyen plus pratique est offert pour gérer les électroménagers. Avec les appareils intelligents, les consommateurs peuvent les contrôler à distance, et ils fonctionnent même automatiquement, basés sur le programme incorporé. Le système de monitoring assure que tous les appareils sont bien fonctionnels, et il garantit la sécurité de la maison. De plus, par la fonction « contrôle des charges dans HEMS », les consommateurs qui suivent la grille tarifaire réglementée (GTR) ou les tarifs différenciés dans le temps (Time-of-Use rates : TOU) peuvent facilement éviter le pic de la charge [11] .

À long terme, d'abord, un HEMS est respectueux de l'environnement. Moins la consommation d'énergie est élevée, moins il y aura d'émission de gaz à effet de serre. Il peut être prévu que le HEMS minimisera grandement la pollution de l'environnement et, dans une certaine mesure, il préservera les ressources non renouvelables à l'avenir. Et puis, il est possible d'influencer la chaîne de valeur d'exigence d'énergie [12] .

2.3 L'importance de passerelle pour HEMS

Le réseau domestique de communication HAN permet l'intégration de plusieurs appareils électroménagers et d'autres appareils intelligents. Les appareils sont normalement fabriqués par diverses compagnies, c'est-à-dire que ceux-ci ne disposeront pas nécessairement du même protocole de communication. Conséquemment, ces appareils ne peuvent pas communiquer entre eux. C'est pour cette raison qu'une passerelle est nécessaire au HEMS afin d'achever l'interopérabilité des éléments du réseau. D'ailleurs, la passerelle est une entité utilisée dans l'infrastructure du réseau; elle fournit une interface entre un réseau local et le réseau Internet (public). Elle contribue également à l'aspect de la

gestion d'énergie. Généralement, une passerelle résidentielle qui est bien développée peut supporter les fonctions suivantes :

- a. L'interconnexion entre l'Internet et le réseau résidentiel et entre les appareils résidentiels.

Les appareils interopérables doivent soutenir le même certificat et la même technologie, car les passerelles dans HAN fonctionnent comme des ponts en vue d'absorber les différents protocoles des réseaux hiérarchiques. Ils permettent aux usagers de contrôler les appareils soit par l'Internet à distance, soit par le réseau local. Sur ce point, les fonctions des passerelles incluent aussi la fonction de la traduction d'une adresse (ex.: IPv4 à IPv6) et un média (ex.: MPEG2/MPEG4) [13].

- b. La gestion et la maintenance du réseau domestique (HAN)

Il importe de s'assurer que le réseau HAN est facile à gérer et maintenir. Dans l'environnement d'une résidence, il faut que le réseau possède une bonne flexibilité d'extension des appareils électroménagers et des capteurs. La passerelle, qui agit comme un coordinateur, vérifie la connexion des appareils automatiquement et périodiquement afin de mettre à jour les appareils connectés. Le signal de chaque appareil vers la passerelle doit être assez fort pour garantir la qualité de communication. Qui plus est, dans l'environnement où coexistent plusieurs systèmes radios, la passerelle de la résidence doit être capable de minimiser l'interférence et l'interférence potentielle avec d'autres systèmes et de choisir un canal approprié automatiquement dans les bandes disponibles.

c. L'interface de gestion à client

Normalement, la passerelle résidentielle offre une interface de contrôle au résident (GUI, Graphical User Interface) [9]. D'un côté, toutes les informations des appareils rassemblés par la passerelle se présentent aux usagers sur GUI. De l'autre côté, elle aide les usagers à envoyer les commandes à l'appareil spécifique. Un navigateur Web est la meilleure façon d'accéder à l'objet à partir des différents environnements.

d. La gestion de dépense et d'énergie

Avec l'implémentation de HEMS, la passerelle résidentielle propose le service de la gestion des charges. La consommation et le coût de consommation sont visibles dans une interface graphique, ce qui favorise le contrôle des appareils par le client dans la maison, dépendamment de ces données. HEMS permet aux consommateurs d'établir une priorité pour tous les appareils dans la maison, d'éviter le pic de consommation et de planifier un calendrier pour surveiller et contrôler les appareils automatiquement. Par conséquent, la passerelle contribue à l'économie d'énergie et optimise la consommation d'énergie dans l'environnement de la résidence [14], [15].

e. La sécurité éprouvée

Il faut que chaque appareil soit associé à l'application de sécurité dans le réseau local; le réseau doit fournir une protection contre les cyberattaques. Les paquets de données doivent être bien protégés. Quand HAN est connecté à l'Internet par la passerelle, il peut survenir quelques problèmes de sécurité. Par cette raison,

afin d'assurer une protection contre les menaces possibles, le pare-feu doit être déployé dans la passerelle [13], [16] .

2.4 Conclusion

Dans ce chapitre, nous avons discuté des difficultés environnementales, liées à la production d'énergie, et à la forte augmentation de la consommation d'énergie résidentielle. Par la suite, nous avons introduit le concept de HEMS qui contribue à l'économie d'électricité, permet aux consommateurs de contrôler les appareils dans le réseau domestique et de gérer l'énergie selon leurs besoins. Un HEMS consiste en un réseau de capteurs, des appareils de mesure, des technologies de communication et un système de gestion de l'énergie. Entre les composants de HEMS, la passerelle est particulièrement importante. Comme une passerelle intègre plusieurs protocoles de communication, elle s'occupe des échanges d'information entre le réseau public et le réseau local. De plus, la passerelle fournit une interface principale pour l'interopérabilité de HEMS; ainsi, toutes les informations de réseau peuvent être présentées au client et les commandes de contrôle peuvent être transportées aux appareils intelligents qui sont connectés.

Chapitre 3 - Comparaison de HomePlug, Wi-Fi, Bluetooth, Z-Wave, ZigBee

Une résidence intelligente aide non seulement à développer la qualité de la vie, mais elle contribue aussi à diminuer la consommation d'énergie. Un réseau local de communication exhaustif offre des fonctions de surveillance et de contrôle des électroménagers, pour mieux échanger l'information avec l'utilité d'électricité. Afin de construire un réseau local pour réaliser ces fonctions, l'un des principaux problèmes qu'il faudra résoudre est la communication dans le réseau local (avec tous ses éléments) et entre celui-ci et le réseau public.

Plusieurs technologies de communication peuvent être choisies, par exemple, Wi-Fi et Ethernet sont les plus populaires présentement; d'ailleurs, il existe des technologies émergentes qui sont plus appropriées au petit réseau : ZigBee, Z-Wave. Chaque protocole comporte des avantages et des inconvénients; ainsi il faut s'assurer de bien choisir une technologie de communication qui favorise à l'interopérabilité et la fiabilité, qui assiste HEMS afin de compléter ses missions efficacement. Toutes les technologies de communication peuvent être classées en deux catégories : filaire (câblée) et sans-fil. Par conséquent, ce chapitre introduit les technologies pour les deux classes respectivement. La section 3.1 parle de la technologie de communication câblée : HomePlug; les sections 3.2, 3.3, 3.4 et 3.5 présentent les technologies sans-fil : Wi-Fi, Bluetooth, Z-Wave, et ZigBee; enfin, la section 3.6 fait une comparaison des technologies.

3.1 Communication par courants porteurs en ligne : HomePlug

HomePlug est une technologie de la communication par courants porteurs en ligne (CPL, ou en anglaise : Power Line Communications). Sa définition est développée par HomePlug alliance qui réunit les chercheurs, les technologistes, les penseurs stratégiques, et bien d'autres, du monde entier, pour se dédier à développer et élargir le marché de CPL connectivité [17]. HomePlug emploie les fils électriques existants dans la maison pour transmettre les données numériques. Afin de construire un réseau, il a le besoin de brancher deux ou plusieurs adaptateurs à la prise électrique, les appareils peuvent alors être connectés au réseau par les adaptateurs [18]. Plusieurs spécifications de HomePlug sont accessibles au marché : HomePlug AV se distingue par son haut débit jusqu'à 200Mbps [19]; HomePlug AV2 est la solution de la haute bande passante qui supporte la transmission de vidéo à haute définition (HD) [20]; et la solution de la maison intelligente avec HomePlug Green PHY qui, elle, a une faible consommation d'énergie [21].

3.1.1 *HomePlug Green PHY*

En général, les principaux avantages de HomePlug sont les suivants : 1) utilisation de câblages électriques qui peuvent être installés à peu près partout dans le monde; 2) capacité d'envoyer les données numériques à haut débit, qualité robuste, ainsi qu'une excellente sécurité; 3) nombre de produits HomePlug qui combinent la mise en réseaux de Wi-Fi, l'élasticité de réseau est donc perfectionnée [22].

Comme HomePlug Green PHY est développé spécifiquement pour une maison intelligente, ce qui est abordé dans ce mémoire représente la technologie de HomePlug. Les produits basés sur le HomePlug Green PHY sont caractérisés par le fait qu'ils ont la plus

vaste couverture et en même temps, ils consomment moins d'énergie. D'ailleurs, HomePlug Green PHY est interopérable avec HomePlug AV, il supporte aussi le réseau Internet (IPv4 et IPv6). Ainsi, il sera facile de créer des applications qui s'intègrent dans un HEMS via un téléphone intelligent ou d'autres interfaces [21]. Cependant, il faut toujours tenir compte de l'interférence électromagnétique et du bruit [8]. Les caractéristiques principales d'HomePlug Green PHY sont présentées par le tableau 3-1.

Tableau 3-1 Les caractéristiques de HomePlug Green PHY [8]

| Caractéristique | Valeur |
|-----------------------------|-----------------------|
| La fréquence d'opération | 2MHz – 30MHz |
| Modulation | OFDM |
| Modulation de sous-porteuse | QPSK |
| Gamme typique | $\approx 100\text{m}$ |
| MAC | CSMA/CA |
| Bande passante | 10Mbps (max) |
| Sécurité | 128-bit AES |

3.1.2 Produits commerciaux disponibles dans la technologie HomePlug

Les produits certifiés de HomePlug peuvent être classés en trois catégories par leurs caractéristiques: adaptateur et routeur; prolongateur de portée Wi-Fi; conception de référence (reference design). La section suivante donne des exemples pour les deux premiers groupes de produit HomePlug.

- 1) AV500 2-Port Powerline Adapter with AC pass Through - TL- PA4020P, par TP-LINK Technologies Co., Ltd.

Cet adaptateur contribue à convertir le réseau électrique en réseau de communication en le brancher dans la prise de courant. Le tableau 3-2 montre les spécifications de cet adaptateur.

Tableau 3-2 Les spécifications de l'adaptateur TL – PA4020P [23]

| AV500 2-Port Powerline Adapter with AC pass Through – TL – PA4020P | |
|---|--|
| Protocole et Standard | HomePlug AV, IEEE802.3, IEEE802.3U |
| Interface | 2x10/100Mbps, Ethernet Ports, Power Socket |
| Type de fiche | EU, UK, FR |
| Indicateur de LED | Power, CPL, Ethernet |
| Consommation de puissance | <3W |
| Gamme | 300m |

- 2) 3-in-1 Wireless HomePlug AV Router – NB318S, par ZyXEL Communication Corp.

Pour bénéficier de la coexistence de la connexion de Courants porteurs en ligne (CPL), Wi-Fi, et Ethernet, ce routeur offre une solution facile pour une résidence numérique. Dans un HAN, le CPL peut servir de base de réseau, et le routeur étend la connexion partout dans la maison. Le tableau 3-3 énumère ses spécifications.

Tableau 3-3 Les spécifications du routeur NB318S [24]

| 3-in-1 Wireless HomePlug AV Router – NB318S | |
|--|--|
| Protocole et Standard | HomePlug AV, Wi-Fi, Ethernet |
| Interface | 3x10/100Mbps Ethernet Ports comme LAN, 1x10/100Mbps Ethernet port comme WAN, Power Socket : 100V-240V, 50Hz-60Hz |
| Débit de donnée | CPL : 200Mbps, Wi-Fi : 108Mbps |
| Sécurité | 128-bit AES de protection de sécurité |

3.2 Wi-Fi

Le Wi-Fi est l'abréviation de Wireless Fidelity, ainsi nommé par l'alliance du début des deux mots : Wi-Fi. Généralement, le Wi-Fi se reporte aux différentes versions du standard IEEE 802.11 (inclut : a/b/g/n. etc.) qui sont les technologies et les protocoles de communication sans fil. Le Wi-Fi est utilisé largement dans tous les domaines : le bureau, la résidence, le restaurant, pour n'en nommer que quelques-uns. En tant que moyen de communication sans-fil populaire, le Wi-Fi est caractérisé par haut débit de transfert de données, ainsi que sa sécurité et stabilité de connexion très élevées. Le débit de données et la fréquence variant entre les différentes versions du standard IEEE 802.11. Tableau 3-4 compare les spécifications des versions du standard IEEE 802.11 a/b/g/n.

Tableau 3-4 Les spécifications du standard IEEE 802.11a/b/g/n [25]

| IEEE 802.11 | Bande de fréquence | Débit de données | Modulation | Gamme (intérieur) |
|-------------|--------------------|------------------|---------------|-------------------|
| 802.11a | 5 GHz | 54 Mbps | OFDM | ≈25m |
| 802.11b | 2.4 GHz | 11Mbps | DSSS/CCK | ≈35m |
| 802.11g | 2.4 GHz | 54Mbps | OFDM/OSSS/CCK | ≈25+m |
| 802.11n | 2.4/5 GHz | >500Mbps | MIMO/OFDM | ≈50+m |

*Notez que le débit de données peut varier selon l'environnement

3.2.1 *Wi-Fi CERTIFIED n*

Le Wi-Fi CERTIFIED n est basé sur le standard 802.11n. Par rapport aux autres versions de standard 802.11 et 802.11n, il opère dans les deux bandes de fréquences 5GHz et 2.4GHz, qui fournissent plus de choix au fabricant pour développer ses applications. L'amélioration la plus significative de 802.11n est l'accès à la technologie MIMO (Multiple-Input Multiple-Output). MIMO emploie de multiples antennes pour envoyer et recevoir de l'information. Cette méthode est bien pratique pour augmenter le débit de données et étendre la gamme de transmission [26]. Pour cette raison, les appareils de Wi-Fi « CERTIFIED n » ont la capacité de libérer jusqu'à plus de cinq fois la vitesse des anciens standards Wi-Fi. Les produits de Wi-Fi « CERTIFIED n » fonctionnent avec les versions antérieures (802.11a/b/g); par conséquent, il n'est pas nécessaire de remplacer les appareils existants.

Les caractéristiques du Wi-Fi « CERTIFIED n » contribuent à trois avantages évidents. Premièrement, son excellente performance et sa vitesse supportent des applications, comme la musique, les jeux et vidéos, qui sont riches en contenu et gourmandes en bande passante,

mais assurent une bonne qualité de multimédia. Par ailleurs, il permet qu'une maison se retrouve complètement dans la couverture d'un signal Wi-Fi fort. De plus, le Wi-Fi « CERTIFIED n » assure un soutien à de nombreux utilisateurs qui se connectent en même temps, sans sacrifier l'intensité du signal [27].

3.2.2 Produits commerciaux disponibles dans la technologie Wi-Fi

De nombreuses compagnies déploient des efforts pour produire des appareils Wi-Fi. Par exemple, Samsung Electronics, Texas Instruments, ZTE Corporation, LG Electronics, D-Link Corporation, etc., et les produits couvrent plusieurs domaines : ordinateur, multimédia, routeur, et maison intelligente, qui ne sont que quelques exemples. La section suivante cite quelques produits pouvant être utilisés dans une maison intelligente.

- 1) Mydlink - Wi-Fi Motion Sensor (DCH-S150), offert par la compagnie D-Link Corporation

Ce capteur de mouvement détecte les mouvements dans la maison où il est installé, et envoie une alerte à un portable intelligent. Il peut être jumelé à la fiche intelligente qui est aussi un produit dans la série de mydlink. Les clients peuvent contrôler tous ces appareils [28]. Le tableau 3-5 donne la liste des spécifications du capteur de mouvement.

Tableau 3-5 Mydlink – Wi-Fi Motion Sensor (DCH-S150)

| Mydlink - Wi-Fi Motion sensor (DCH-S150) | |
|--|---|
| Standard | IEEE 802.11n |
| Fréquence | 2.4GHz à 2.4835GHz |
| Gamme | 7.9248m gamme de détection |
| Puissance | Entrée : 110 à 125V AC, 50/60Hz |
| Fonctionnalités avancées | Support une application pour iOS et Android |

- 2) Lyric Round Wi-Fi Thermostat – Second Generation (RCH9310WF), par Honeywell International Inc.

Le thermostat Lyric peut être contrôlé de n'importe où, à l'aide d'un téléphone intelligent; il fonctionne aussi en se basant automatiquement sur les habitudes des clients. Les clients sont capables de consulter les prévisions météorologiques des 6 à 12 prochaines heures, la température et l'humidité réelle dans la maison. Toutes ces informations peuvent servir aux clients afin de dresser un horaire de la journée et que ce thermostat soit réglé selon un environnement confortable, automatiquement. Ce thermostat utilise l'emplacement du téléphone intelligent pour se changer en mode « veille », qui consomme moins d'énergie, lorsque le client est parti à une certaine distance de la résidence. Le tableau 3-6 présente les spécifications de ce thermostat [29].

Tableau 3-6 Thermostat Lyric Round Wi-Fi

| Thermostat Lyric Round Wi-Fi – Second Generation (RCH9310WF) | |
|--|----------------|
| Protocole de communication | 802.11 b/g |
| Bande de fréquences | 2.4GHz |
| Voltage | 24V |
| Système d'application | Android et iOS |

3.3 Bluetooth

Le Bluetooth est basé sur le système radio sans fil conçu pour les appareils communicants à courtes distances et à un coût peu élevé [30]. Il opère dans la fréquence de 2.4GHz et les appareils peuvent partager les données à moins de dix mètres à la vitesse

720kbps de capacité [31]. Deux topologies fondamentales sont possibles avec Bluetooth : piconet et scatternet.

Un réseau de piconet est formé par un dispositif Bluetooth qui fait office de maître dans le réseau, et un ou plusieurs dispositifs Bluetooth font office de serveurs (le rôle peut être inversé). Un dispositif maître peut être connecté avec sept dispositifs serveurs dans un piconet, et le maître décide du temps de synchronisation pour tous les dispositifs. Les serveurs communiquent avec leur maître selon le mode de point à point, tandis que le maître est capable de choisir la mode de transmission, soit point à point, soit point à multipoint. Si un des serveurs d'un piconet agit également comme maître d'un autre piconet, ils forment un scatternet [32].

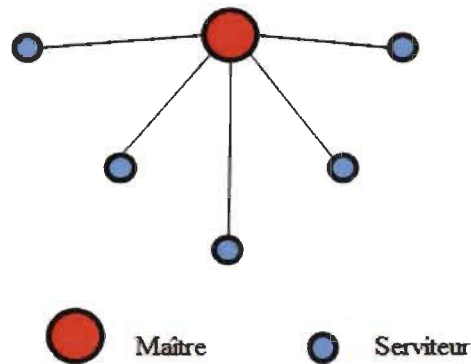


Figure 3.1 Topologie de piconet

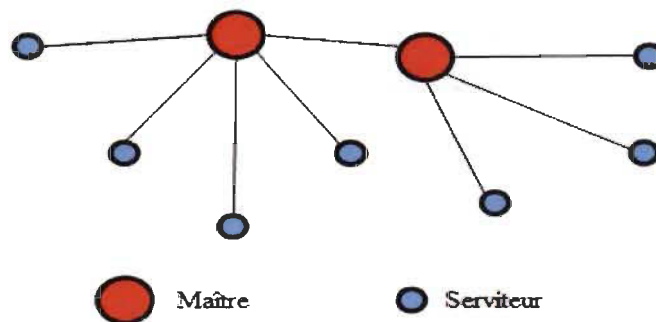


Figure 3.2 Topologie de scatternet

3.3.1 Bluetooth 4.0 + (ou Bluetooth Smart; Bluetooth Low Energy)

Avec la tendance concernant la protection de l'environnement et l'économie d'énergie, le nouveau Bluetooth technologie - BLE (Bluetooth Low Energy) est apparu comme un besoin, en 2010. Par rapport aux versions ultérieures de Bluetooth, la faible consommation d'énergie de BLE le rend idéal pour les dispositifs qui sont équipés de sources d'énergie telles que la pile bouton, qui fonctionnent sur une longue période. Le tableau 3-7 établit une comparaison entre le Bluetooth classique et BLE. Le BLE réduisait beaucoup la consommation d'électricité, il améliorerait ainsi la sécurité. Cette nouvelle technologie est mieux adaptée au marché des résidences intelligentes, des dispositifs portables pour la santé ou du système sur Internet of Thing, etc. Il convient de noter que BLE n'est pas compatible avec les versions ultérieures.

Tableau 3-7 Comparaison entre le Bluetooth classique et BLE [33][34]

| Caractéristiques | Bluetooth Classique | BLE |
|---------------------------|---------------------|--------------|
| La fréquence d'opération | 2.4GHz | 2.4GHz |
| Taux de données | 1-3 Mbit/s | 1Mbit/s |
| Débit | 0.7-2.1Mbit/s | 0.27Mbit/s |
| Gamme typique | 10m/100m | >100m |
| Consommation d'électrique | ≈1W | 0.01W à 0.5W |
| Sécurité | 56 à 128 bit | 128-bit AES |

3.3.2 Produits commerciaux disponibles dans la technologie Bluetooth

L'utilisation de la technologie Bluetooth s'étend à plusieurs domaines et des milliards de produits Bluetooth existent sur marché. Au cours des dernières années, Bluetooth est apparu sur le téléphone, la voiture, l'ordinateur, la télévision, la tablette, etc. Ci-dessous,

nous présentons des exemples des produits Bluetooth qui sont utilisés dans une résidence, ce sont des produits qui peuvent contribuer au HEMS.

1) Robosmart Wireless LED Light Bulb, par Smartbotics Inc.

L'ampoule Robosmart LED est une ampoule innovante contrôlée par un Bluetooth connecté à un téléphone intelligent. L'application dans un téléphone intelligent permet au client de configurer la luminosité, le nom des ampoules, le nom des groupes, les calendriers d'économies et d'utilisation d'énergie.

Tableau 3-8 Spécifications des ampoules D.E.K. de Robosmart [35]

| Robosmart Wireless LED Light Bulb | |
|-----------------------------------|----------------|
| Connexion | Bluetooth 4.0 |
| Rendement lumineux | 60W |
| Consommation d'électrique | 12.7W |
| Gamme | ≈60m |
| Système d'application | iOS et Android |

2) EVE Energy switch & meter, par la compagnie Elgato

Ce produit sert comme interrupteur et capteur de puissance. Il a la capacité de communiquer avec un téléphone ou une tablette d'iOS par Bluetooth. Le dispositif qui s'insère dans l'interrupteur peut être contrôlé par téléphone ou tablette, et même la consommation d'électrique est affichée sur un beau graphique [36].

Tableau 3-9 Caractéristiques d'EVE Energy switch & meter

| EVE Energy Switch & Meter | |
|----------------------------|--|
| Connexion | Bluetooth 4.0 |
| Entrée/Sortie | AC 100-240V, 50/60Hz |
| Consommation d'électricité | Nord-américain : 1800W Europe : 2500W |
| Gamme | ≈50m |
| Système d'application | iOS 9.1 ou ultérieur |

3.4 Z-Wave

Z-Wave est une technologie de communication sans-fil qui est appropriée pour une communication sur des courtes distances, elle est axée sur l'automatisation de la résidence et l'environnement commercial. Elle est à l'avant-garde des technologies pour la gestion d'énergie intelligente.

3.4.1 Les caractéristiques de Z-Wave

La pile de protocole Z-Wave consiste simplement en quatre couches (figure 3-3) : (1) la couche de PHY/MAC, qui contrôle le média de RF; (2) la couche de transport, qui est responsable de transmettre et de recevoir les trames; (3) la couche de routage, qui contrôle le routage des trames d'un nœud à un autre; et (4) la couche d'application, qui exécute les commandes [37].

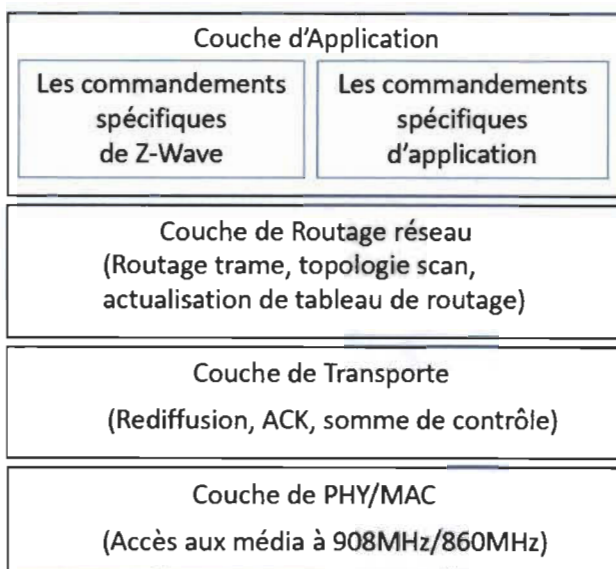


Figure 3.3 La structure du protocole de Z-Wave [38]

Le protocole radio Z-Wave fonctionne sur la bande de 908.42 MHz aux États-Unis et au Canada. Avec une faible bande passante cette technologie assure une communication sans-fil stable et moins de consommation d'énergie. Deux appareils Z-Wave sont capables de communiquer ensemble à distance, soit jusqu'à 30m avec une vitesse de 40 kbps. Un réseau de Z-Wave emploie une topologie maillée; chaque réseau peut admettre un maximum de 232 nœuds. On identifie dans le réseau deux groupes : contrôleur et serviteur. Les appareils de contrôle sont les nœuds qui envoient les commandes initiales aux autres nœuds, alors que les nœuds serveurs exécutent les opérations demandées. Autrement dit, le serviteur transmet les commandes aux autres nœuds, de manière à permettre au contrôleur de communiquer avec les serveurs indirectement [39]. La figure 3-4 donne un exemple de la topologie de Z-Wave. Un numéro identifiant unique de 32 bit, appelé Home ID, est attribué d'avance à chaque nœud contrôleur, et le contrôleur attribue une adresse de 8 bits (dit à Nœud ID) à chaque nœud serviteur qui se trouve dans le réseau [38].

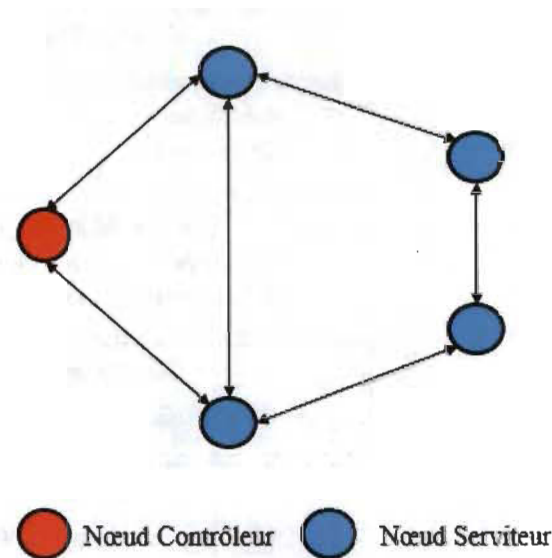


Figure 3.4 Topologie de Z-Wave

En résumé, Z-Wave est une technologie de communication fiable, à faible puissance, à un coût peu élevé, qui vise la communication à courte distance, particulièrement à l'application de contrôle à distance dans une résidence. Grâce à la pile de protocole légère et au format des trames à compression, Z-Wave a une consommation extrêmement faible d'énergie; la topologie Mesh permet aux appareils de s'échanger l'information; puisque Z-Wave n'est pas conçu pour transférer une grande quantité de données, sa bande et portée lui permettent de fonctionner très faible puissance; Z-Wave a moins de possibilité d'interférence avec les autres dispositifs sans-fil [37].

3.4.2 Produits commerciaux disponibles dans la technologie Z-Wave

Selon les données recueillies à propos de Z-Wave alliance, on compte à peu près 40 millions de produits Z-Wave dans le monde, concerne différents sortes d'électroménagers et appareils. Les produits se retrouvent principalement dans les catégories des dispositifs

d'éclairage, serrures, capteurs et thermostats. Nous citerons ci-dessous quelques exemples de Z-Wave thermostat et de Z-Wave contrôleur avec interrupteur.

1) Thermostat Vision Pro Z-Wave, par Honeywell

Le thermostat Vision pro Z-Wave est un thermostat programmable. Il offre un écran tactile à interaction, une horloge en temps-réel pour un contrôle aisé du client. Ce thermostat donne à son propriétaire la capacité de gérer la température de la maison à distance, et d'atteindre le résultat visé d'économie d'énergie. Le programme personnalisé porte le propriétaire à vivre dans un environnement confortable, qui correspond à ses habitudes [40].

Tableau 3-10 Spécifications du thermostat Vision Pro Z-Wave

| Vision Pro Z-Wave thermostat | |
|------------------------------|---------------------------------|
| Protocole de communication | Z-Wave |
| Puissance | 24VAC (fourni par HVAC Système) |
| Capacité de Programme | 7 jours |
| Économies d'énergie | ≈33% chaque année |

2) Deux boutons contrôleurs avec interrupteur, VRCS2-MRZ, par LEVITON

Ce produit est capable de transformer un interrupteur ordinaire en un interrupteur de télécommande pour contrôler l'éclairage à distance. Deux charges peuvent être contrôlées par un interrupteur. Il répond aussi aux commandes d'autre contrôleur Z-Wave [41].

Tableau 3-11 2-boutons contrôleur avec interrupteur, VRCS2-MRZ - LEVITON

| 2-boutons contrôleur avec interrupteur, VRCS2-MRZ - LEVITON | |
|---|-------------------------------------|
| Protocole de communication | Z-Wave |
| Tension | 120VAC/240VAC |
| Puissance (courant) | 1000W (1000VA) |
| Fréquence | 50/60Hz |
| Compatibilité | Autres Z-Wave dispositifs autorisés |

3.5 ZigBee

Le protocole ZigBee est basé sur le standard IEEE 802.15.4 Personal-Area Network (PAN), développé par l'Alliance ZigBee, utilisé pour le monitoring et l'application de contrôle [8]. Il ressemble au protocole Z-Wave, car c'est une technologie de communication sans-fil qui a un faible débit de donnée et une faible consommation d'énergie. La pile du protocole ZigBee se compose de quatre couches : couche physique (PHY), couche de contrôle d'accès au support (MAC), couche de réseau (NWK), et couche d'application (APL). Le standard IEEE 802.15.4 offre le niveau 1 (PHY) et le niveau 2 (MAC) de la pile de protocole, le reste des deux niveaux sont donc définis par ZigBee. La figure 3-5 illustre la structure du protocole ZigBee [42].

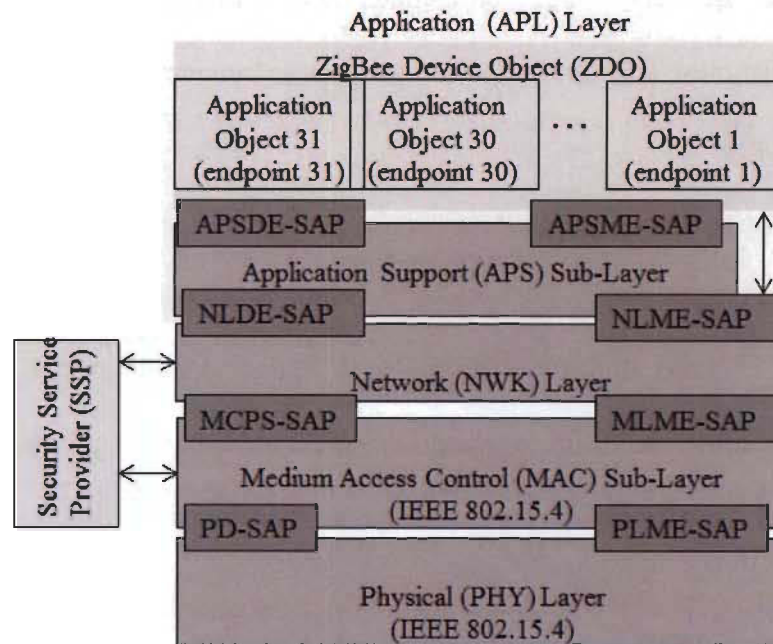


Figure 3.5 La structure du protocole ZigBee [42]

3.5.1 Caractéristiques du protocole ZigBee

Les trois premières couches (PHY, MAC, et NWK) s'occupent de la création et de l'entretien du réseau, elles garantissent théoriquement le débit de transmission de données jusqu'à 250kbps [39]. La couche d'application est responsable de la communication entre les dispositifs, par exemple, pour envoyer une commande ou demander une information [43]. Le réseau de ZigBee supporte différentes topologies, le réseau en étoile, en pair à pair, et maillé. Trois types de nœuds peuvent exister dans un réseau ZigBee: 1) coordinateur, qui s'occupe de la création et du contrôle de réseau. Les informations des autres nœuds (adresse, données recueillies...) sont stockées et échangées par le coordinateur; 2) routeur, qui n'est pas un élément essentiel pour le réseau ZigBee, car un routeur est capable de se connecter au coordinateur et aux autres routeurs, ainsi qu'aux appareils. Il sert à transmettre les informations reçues afin d'étendre le réseau. Par rapport au coordinateur, le routeur n'a pas la capacité de créer un réseau et leurs principales fonctions sont similaires; et 3)

appareil qui répond aux commandes de coordinateur, transmet ou envoie les données au capteur, c'est-à-dire un appareil qui doit se connecter soit avec le coordinateur soit avec routeur [39], [43].

ZigBee fonctionne sur la bande de 2,4GHz (au niveau mondial), de 868MHz (en Europe), de 915MHz (aux les États-Unis), et le taux de transmission de données maximal est de 250kbps, 40kbps, et 20kbps [44]. Malgré la puissance typique de ZigBee, soit 1mW, ce qui limite sa distance d'utilisation, ce n'est pas un problème, car son objectif est le HEMS. De plus amples caractéristiques générales de ZigBee sont montrées au tableau 3-12.

Tableau 3-12 Les caractéristiques de ZigBee[32]

| Les Caractéristiques de ZigBee | |
|---------------------------------|-----------------------------|
| Bande de fréquence | 2.4GHz; 868/915MHz |
| Débit de transmission de donnée | 250kbps |
| Distance | 10 – 100+m |
| Topologie | Étoile, Pair à Pair, Maillé |
| MAC | CSMA-CA |
| Puissance | -25 – 0dBm |
| Nœuds | 65,535 |

CSMA-CA : un mécanisme d'accès aux canaux pour améliorer le débit de réseau et minimiser le délai de transmission

Dans un scénario HEMS, ZigBee possède à la fois de nombreux avantages et quelques inconvénients. Les principaux points sont énumérés ci-dessous.

Les avantages d'adopter ZigBee à HEMS :

- Grâce au simple protocole, ZigBee est facile à mettre à œuvre [45];
- Faible consommation d'énergie, ce qui permet aux appareils qui ont sources par batterie: capteur, moniteur, contrôleur, etc., de durer plus longtemps [46];

- Plus que 6000 nœuds peuvent coexister dans un réseau ZigBee, ce qui suffit à une grande maison avec plein d'appareil et capteurs ZigBee;
- Le temps de latence de ZigBee est court, environ 15 ms à 30 ms, ce qui garantit la transmission de donnée pour une gestion en temps-réel [46];
- Le protocole ZigBee est capable d'organiser (reconfigurer) le réseau automatiquement [8].

En revanche, il conviendrait de considérer les trois désavantages suivants de ZigBee:

- Le manque de soutien du protocole Internet dans sa version antérieure. En fait, cette caractéristique est définie par ZigBee Smart Energy Profile 2 (SEP2) lequel est exposé au chapitre IV. Par conséquent, un réseau ZigBee doit s'appuyer sur une passerelle spécifique à l'accès d'Internet, qui augmente le coût d'implantation du réseau local [8].
- La majorité des appareils ZigBee ne sont pas compatibles avec les autres protocoles de communication [46].
- ZigBee a un faible débit de transmission de donnée (maximum 250 Kbps avec 2,4GHz). Par contre, c'est un élément qui fait que ZigBee comporte une faible consommation d'énergie [8].

3.5.2 *Produits commerciaux disponibles dans la technologie ZigBee*

1) Capteur Wireless Contact Sensor (MCT-340 SMA), par Visonic

Ce capteur ZigBee est employé pour détecter l'état d'ouverture et de fermeture des portes ou des fenêtres dans la maison. Les informations de surveillance sont envoyées

périodiquement pour informer le récepteur des activités de chaque unité dans le système [47].

Tableau 3-13 Spécification du capteur MCT-340 SMA

| Wireless Contact Sensor (MCT-340 SMA) | |
|---------------------------------------|------------------------------------|
| Protocole de communication | ZigBee Home Automation 1.2 |
| Fréquence | 2.4 GHz |
| Alimentation | 3V (deux piles 1.5V) |
| Capacité de pile | 230mAh |
| Durée de vie de pile | En années (en utilisation normale) |

2) Contrôleur “Wireless Area Controller (WAC50)”, par Daintree Networks

Ce contrôleur, de Daintree Networks, est conçu pour fournir les commandes dans un réseau local, à de nombreux des appareils intelligents et interopérables qui sont compatibles avec les produits de Daintree. Par la technologie ZigBee, ce contrôleur communique avec d'autres appareils ZigBee compatibles, comme des capteurs et, des interrupteurs, afin de transformer un simple contrôle local en une solution complexe de gestion d'énergie [48].

Tableau 3-14 Spécification du contrôleur Wireless Area Controller (WAC50)

| Wireless Area Controller (WAC50) | |
|----------------------------------|--------------------------|
| Protocole de communication | ZigBee et Ethernet |
| Alimentation | 5V DC 1.5A |
| RF | 2.4GHz 100mW (+20dBm) |
| Consommation d'énergie | 2.8W |

3.6 Comparaison des protocoles de communication pour le système de gestion locale

La conception et les exigences de HEMS conduisent à une considération spécifique quant aux technologies de communication. Peu importe si la communication se fait par un courant porteur en ligne (CPL) ou par les protocoles sans-fil, ils ont leurs propres caractéristiques et avantages, qui conviennent à divers scénarios. Une comparaison de ces technologies est abordée dans cette section, afin d'aider le consommateur à faire des choix judicieux quant à la mise en place de HEMS. Les caractéristiques des technologies sont indiquées dans le tableau 3-15 et les descriptions concrètes sont présentées dans les paragraphes suivants.

Tableau 3-15 Comparaison des technologies de communication

| | HomePlug | Wi-Fi | Bluetooth | Z-Wave | ZigBee |
|-------------------|----------|--------------------|---------------|-----------------|-------------------------------|
| Standard/Protocol | IEEE AV | IEEE a/b/g/n | IEEE 802.15.1 | Z-Wave | ZigBee |
| Fréquence | 2-28MHz | 2,4GHz /5GHz | 2,4GHz | 908M/ 860MHz | 868MHz/ 915MHz/ 2,4GHz |
| Débit de donnée | 200Mbps | 54Mbps/ 100Mbps | 1Mbps | 40Kbps | 20Kbps/ 40Kbps/ 250Kbps |
| Portée (distance) | 100-300m | 250m | 100m | 30m | 10-100+m |
| Topologie | / | Étoile | Étoile | Maillé | Maillé/Étoile |
| Nœuds | / | 2007 [32] | 8 | 232 | 6500 |

HomePlug est la technologie de CPL la plus réalisable pour HEMS, elle est supérieure aux autres, à cause de la possibilité d'utiliser les lignes électriques de la maison pour la communication. Grâce à cet avantage, un réseau de HomePlug peut éviter les coûts additionnels reliés à l'installation de fils supplémentaires. En outre, HomePlug implique un haut débit de donnée, ce qui permet de transmettre rapidement de grandes quantités de fichiers, par exemple envoyer les données vers une télévision à haute définition.

Par rapport à HomePlug, même si les technologies de communication sans-fil engendrent plus de flexibilité, d'interopérabilité, et moins de consommation d'énergie, il y a d'autres problèmes qui doivent également être pris en considération. La comparaison entre les technologies sans-fil va concerner d'autres aspects : 1) canaux de radio et fréquence; 2) temps et distance de transmission; 3) taille de réseau; 4) sécurité; et 5) consommation d'énergie.

1) Canaux de radio et fréquence

Les protocoles Wi-Fi, Bluetooth, et ZigBee emploient les technologies à « spectre étalé » dans la bande 2.4 GHz. Les technologies « spectre étalé » se composent de FHSS (Frequency-Hopping Spread Spectrum) et DSSS (Direct Sequence Spread Spectrum). Le Wi-Fi utilise le DSSS à 14 canaux RF et 22 MHz de bande passante. Le Bluetooth utilise le FHSS à 79 canaux et 1 MHz de bande passante, tandis que ZigBee emploie le DSSS à 16 canaux et 2 MHz de bande passante. Cependant, Z-Wave opère sur la bande 908.42 MHz aux États-Unis et au Canada, 868.42 MHz en Europe; il a seulement un canal et utilise le codage Manchester [32], [49].

2) Débit de données et distance de transmission

Le débit de donnée est un autre facteur de performance à considérer pour les technologies qui s'implantent dans un HEMS. Ce paragraphe aborde cet aspect à partir de la technologie qui a le plus haut débit de données et dont la distance de transmission est variée, selon l'environnement. Le Wi-Fi supporte le débit de communication de 2Mbps à 600Mbps, la couverture est jusqu'à 100 m. Le Bluetooth comporte le maximum de débit de donnée, soit 720Kbps dans 10 m. ZigBee a différents débits de transmission pour les deux fréquences. Il est limité à 20Kbps avec la portée maximale de 75 m en Europe, en attendant, aux États-Unis, le débit de données est jusqu'à 250Kbps pour approximativement 100 m. Z-Wave transmet les données à une vitesse qui se situe entre 9.6Kbps et 40Kbps avec une portée à environ 30 m [31], [45], [49].

3) Taille de réseau

Tous les protocoles sont aptes à construire un réseau complexe à partir des éléments fondamentaux. Le protocole ZigBee soutient le réseau maillé, par rapport aux trois autres, il a le nombre le plus élevé de nœuds maximaux qui dépassent 65 000. Le deuxième est le réseau Wi-Fi, avec 2007 nœuds maximaux. Comparativement au ZigBee, le réseau Z-Wave à maillé, avec un maximum de 232 dispositifs de capacité à contenir. En dernière place, on retrouve un piconet de réseau Bluetooth construit, avec 8 nœuds [32], [49].

4) Sécurité

La sécurité est le quatrième élément considéré pour analyser la performance d'un protocole. Pour instant, on compte deux méthodes populaires de cryptage employées dans le Wi-Fi, sous WPA2-PSK(AES). Le WPA2-PSK(AES) est la

version améliorée de WPA-PSK(TKIP), de ce fait, WPA-PSK(TKIP) est remplacé par WPA2-PSK(AES) dans les nouveaux produits de Wi-Fi. Bluetooth utilise le chiffrement EO par flot pour densifier les données de transmission. La puce originale de Z-Wave adopte l'algorithme triple DES (triple data encryption algorithm) qui est considérée comme étant non sécuritaire, pour la nouvelle version, et qui utilise la même technologie que ZigBee, AES-128 (Advanced Encryption Standard) qui est un chiffrement par bloc à haute sécurité [8], [32], [49].

5) Consommation d'énergie

Le dernier point, mais non le moindre, est l'énergie consommée pendant la transmission de donnée par les protocoles qui doivent être pris en compte, surtout pour l'importance des dispositifs qui changent par batterie. Le Wi-Fi est conçu pour la communication à longue distance et les dispositifs ont une source substantielle. En conséquence, une communication par Wi-Fi consomme plus d'énergie que les trois autres protocoles. Par contre, comme Bluetooth, Z-Wave, et ZigBee sont destinés aux dispositifs portables qui ont des sources limitées, ils ont une faible de consommation d'énergie [32].

En résumé, les différentes caractéristiques des cinq protocoles les rendent appropriés selon diverses situations. Dans notre cas, l'objectif de recherche est la mise en place des technologies de communication à HEM. Selon les performances des technologies qui ont été abordées ci-dessus, ZigBee pourrait être le meilleur choix pour le contrôle et la surveillance dans HEMS, car il est relativement facile à implanter et à développer; la faible consommation d'énergie convient aux dispositifs avec une source de batterie (capteur);

d'ailleurs, il soutient une grande quantité de nœuds; de plus, il est plus populaire dans les produits commerciaux.

3.7 Conclusion

À propos de la connectivité de HEMS, il existe plusieurs technologies et standards de communication. Ce chapitre présente tout d'abord respectivement les caractéristiques des technologies HomePlug, Wi-Fi, Bluetooth, Z-Wave et ZigBee. Ensuite, nous avons comparé ces cinq technologies sur la fréquence de communication, le débit de donnée, la distance approximative de transmission, la topologie, le nombre de nœuds permettant de se connecter et la sécurité. En considérant que ce travail vise le réseau domestique, ZigBee comporte plus d'avantages par rapport aux autres technologies mentionnées. ZigBee peut être utilisé pour construire un grand réseau avec environ 6 000 nœuds, car même s'il comporte une limite sur la distance de transmission, la distance qu'il soutient satisfait le besoin dans une résidence. En plus, ZigBee consomme moins d'énergie que les autres.

Chapitre 4 - Smart Energy Profile 2.0 (SEP2.0)

Comme mentionné dans les chapitres précédents, l'interopérabilité pour un réseau intelligent est essentielle. Ainsi, « ZigBee Smart Energy » est le principal standard du monde en ce qui concerne la gestion d'énergie. Il vise les produits interopérables qui servent à surveiller, contrôler, informer et distribuer l'énergie. Ce standard public est employé dans les applications de réseau intelligent comme le mesurage et HAN [50]. La mise en œuvre des versions « ZigBee Smart Energy 1.x (SEP1) » a été essentiellement motivée par l'intérêt sur l'implémentation d'un réseau intelligent avec gestion de la demande et contrôle direct de charges (Demand Response and Load Control protocol - DRLC). Cependant, ZigBee Smart Energy a beaucoup évolué dans la nouvelle version « ZigBee Smart Energy 2.0 (SEP2.0) », et il a été identifié comme le standard d'interopérabilité comme un réseau intelligent par le « National Institute of Standards and Technology (NIST) ». La nouvelle version est essentiellement améliorée sur les aspects de structure, mécanisme de sécurité, et efficacité de communication [51]. Dans ce chapitre, une description de SEP2.0 permettra de préciser ses fonctions et avantages dans le scénario de HEMS.

4.1 Introduction au SEP2.0

Par rapport aux versions précédentes, ZigBee Smart Energy profile 2.0 (SEP2.0) se concentre plus sur la couche d'application, il apporte un standard ouvert à l'utilité, aux fournisseurs tiers de service, et au consommateur pour leur interopérabilité d'information

sans risque. Le SEP2.0 est construit au-dessus du protocole IP et l'agnostique de couche PHY/MAC ce qui permet l'interopérabilité entre les technologies multiples de communication, telles que IEEE 802.15.4, courants porteurs en ligne, et Wi-Fi. Une comparaison des structures des protocoles ZigBee, modèle OSI, SEP1.x et SEP2.0 est affichée à la figure 4-1.

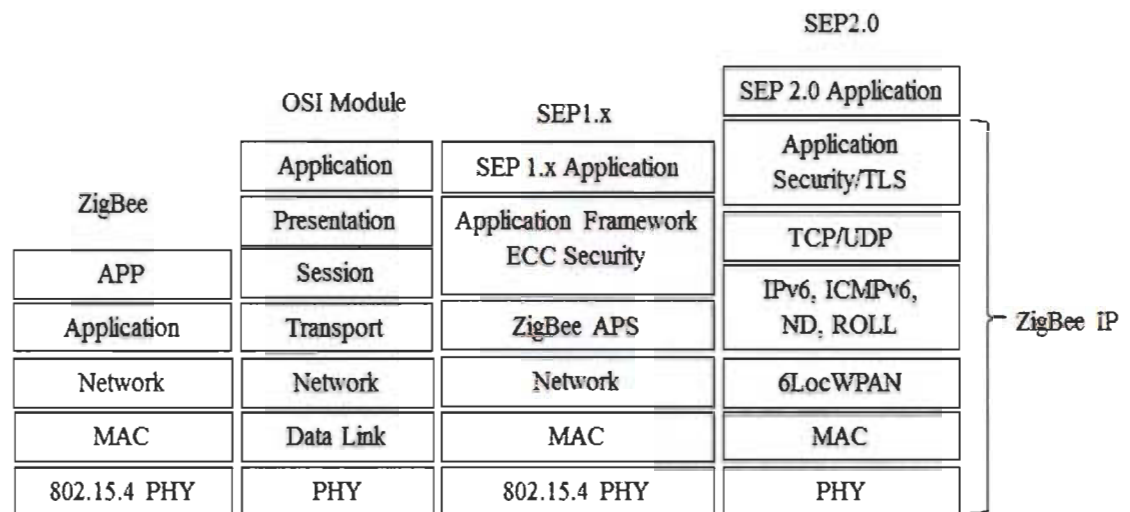


Figure 4.1 Les structures des protocoles ZigBee, modèle OSI, SEP1.x et SEP2.0

La figure 4-1 témoigne que le standard SEP1.x est basé sur le cadre de ZigBee; par contre, SEP2.0 est développé en se basant sur la technologie qui s'appelle ZigBee IP. Cette technologie est conçue spécifiquement pour fournir une pile de protocoles IPv6 qui peut opérer sur les appareils qui consomment moins d'énergie, ainsi que pour soutenir les besoins de SEP2.0 [51]. Le standard SEP2.0 repose sur les technologies ouvertes, telles que REST/HTTP et XML. L'architecture de SEP2.0 adopte le style REST (Representational state transfer); les éléments, attributs, et ressources de SEP2.0 sont décrits par le standard XSD (XML Schema Definition) et le module CIM (Common Information Model).

Les nouveaux types d'appareils que SEP2.0 soutient comprennent les terminaux de paiement, les ESIs (Energy Services Interface), les onduleurs, les appareils de contrôle de charge, les véhicules *Plug-In*, etc. Il convient de noter que tous les appareils SEP2.0 doivent comporter un certificat spécifique [50], [52]. Dans la spécification de SEP2.0, la notion de « *function-sets* », qui est la collection des caractéristiques et des commandes, a été définie [53]. En raison de SEP2.0 qui est développé pour satisfaire les besoins de communication avec les appareils dans un HEMS, diverses séries de fonctions (*function-sets*) ont été ajoutées, lesquelles comprennent [53]:

- Réponse aux demandes et le contrôle de charge;
- Messagerie dirigée;
- Coût;
- Paiement d'avance;
- Information de Compteur et de consommation d'énergie;
- Surveillance;
- Véhicule électrique et *Plug-in*;
- Gestion de ressource d'énergie distribuée;
- Facturation;
- Téléchargement du dossier.

4.2 Architecture de SEP2.0 et les technologies supportées dans la couche d'application

Ce protocole d'application est conçu pour l'implémentation de l'architecture REST, qui est largement utilisée à déployer un service de web sur HTTP (Hypertext Transfer Protocol).

Le protocole HTTP définit la communication entre les différentes parties du web, alors que l'échange repose sur les requêtes client et serveur. Le serveur est celui qui expose une ressource à un client et le client est celui qui a des interactions avec la ressource. Ce module client/serveur contient les opérations de GET, HEAD, PUT, POST, et DELETE vers les ressources. Les opérations sont encodées par XML et/ou EXI, les ressources réfèrent aux objets qui opèrent via l'interface de RESTful et qui sont capables d'utiliser l'adressage URI (Uniform Resource Identifier). Considérant la flexibilité et l'extensibilité de réseau, les URI des clients ne sont pas fixés pour tous les serveurs. Si un URI répond à un résultat inattendu, le client doit recommencer à chercher un nouvel URI. D'ailleurs, l'interface de RESTful est définie par WADL (Web Application Description Language) [51].

Parce que HTTP prend TCP comme protocole de transport, TCP fournit l'assurance de distribution et gère la session de fenêtrage [51]. Afin de permettre aux appareils dans le réseau d'être trouvés automatiquement par les serveurs, le service xmDNS est utilisé par la porte de TCP. Par exemple, la passerelle peut découvrir automatiquement tous les appareils actuels et nouveaux dans le réseau local [54].

4.3 La sécurité

La sécurité dans la couche d'application vise à assurer la sécurité de transaction entre les clients et les serveurs, aussi que l'intégrité des données. Cet objectif est accompli à la base de l'utilisation de TLS (Transport Layer Security). Le protocole TLS est composé de deux parties : TLS record et TLS Handshake. TLS record se situe en haut du protocole TCP, et en bas de TLS Handshake; il s'occupe de crypter les informations et de les envoyer, ainsi que de décrypter les informations reçues.

Avant qu'un nouveau nœud participe à un réseau de SEP 2.0, il faut que le nœud ait obtenu l'authentification et l'autorisation. En dehors de cela, peut-être qu'un enregistrement est nécessaire, ce qui est fourni par l'utilité ou tiers fournisseur de service. De plus, il supporte aussi la clé publique d'infrastructure ECC (Elliptic Curve Cryptography) [55]. Dans le système de gestion du certificat de SEP 2.0, il y a une clé publique infrastructure (public key infrastructure, PKI), qui est installée lors de l'installation de l'application. L'utilisation du certificat d'ECC sert à une identité pour chaque appareil d'énergie intelligente de ZigBee, et l'aide à joindre le réseau. Le certificat de test est accessible sur le site Web de Certicom (<https://www.certicom.com/>).

4.4 Ressource

Les ressources de SEP 2.0 sont créées par UML (Unified Modeling Language), et classifiées selon trois grandes catégories : ressource d'appui; ressource communes et ressource d'énergie intelligente.

Les ressources d'appui sont utilisées pour l'opération d'information des appareils. Elles possèdent des fonctions qui incluent les capacités de l'appareil, l'interface d'appareil, l'interface d'information, l'attribution de ressources, le mécanisme de souscription et la notification de ressources, ainsi que la réponse.

Les ressources communes fournissent les fonctions générales, sans rapport avec un domaine spécifique. Les fonctions suivant sont incluses : le temps, les informations d'appareil, le statut de source, le statut de réseau, la liste de journal des événements, la configuration d'appareil, le téléchargement de fichier.

Les ressources d'énergie intelligente sont visées selon le domaine d'énergie intelligent. Dans cette partie, il définit les fonctions de : fonctionnements communs pour l'application d'énergie intelligente, contrôle direct de charges (DRLC), mesurage, la tarification, la messagerie, la facturation, le paiement anticipé, la réservation de flux d'énergie, et DER (Distributed Energy Resources) [51].

La fonctionnalité de DRLC est ciblée par les dispositifs du client (ex.: thermostat, chauffe-eau) et le serveur (ex.: Energy Services Interface). Avec cette fonctionnalité, l'événement de DRLC pourrait être programmé selon les dispositifs spécifiques, destiné à réduire la puissance totale de la charge pendant certaines périodes. Plusieurs méthodes sont disponibles pour le contrôle de charge, y compris la valeur de température définie et compensation, soit le niveau critique et le cycle de travail. Les dispositifs ont la capacité d'être groupés afin de synchroniser leurs actions. Bref, la fonction de DRLC est utilisée essentiellement pour éviter le pic de la demande vue par l'utilité, et pour aider les clients à économiser la dépense d'énergie.

4.5 Les fonctions principales

Comme il a été mentionné dans la description ci-dessus, SEP2.0 soutient plusieurs fonctions, chaque appareil de SEP2.0 dispose d'un ou de plusieurs types de fonctionnement. Afin de répondre aux besoins de HEMS et de gérer la consommation d'énergie de la maison, quelques fonctions cœurs des ressources d'énergie intelligente devraient être remarquées. Les fonctionnalités les plus importantes sont le mesurage d'énergie, la tarification et le contrôle direct de charges (DRLC - Demand-Response Load Control).

La fonctionnalité du mesurage d'énergie permet aux consommateurs d'accéder aux informations des multiples produits, y compris l'électricité, le gaz, l'eau. Ces informations proviennent du compteur intelligent ou de l'appareil qui a la fonction de mesurage, par exemple les appareils intelligents. Un dispositif d'affichage, un portable intelligent ou une tablette, qui sont équipés de l'application SEP2.0 peuvent présenter les informations de consommation d'énergie en temps-réel. Grâce à cette fonctionnalité, divers types de mesures sont disponibles, tels que le profil de la charge, ou la sommation de l'utilisation. Toutes les informations peuvent être enregistrées.

Figure 4.2 présente à titre d'exemple le processus d'échange entre un appareil client (ex : In-Premises Display) qui demande de lire les informations de mesurage d'un usage. Le client demande d'abord une liste d'usage (*UsagePoint List*) qui comprend les adresses des compteurs locaux, un usage peut-être lié avec plusieurs compteurs (*Meter Readings*), et puis le serveur de compteur répond avec la liste. Ensuite, le client demande de chercher une liste de « *Meter Readings* » qui inclut tous les types d'attributs (*ReadingType*) d'un compteur spécifique. Après le serveur répond avec la liste de « *Meter Reading* », le client demande ensuite la liste de « *ReadingType* » qui comprend les types par exemple la somme de délivrée, la somme de reçue, et la demande instantanée. Pour l'étape finale, le client reçoit la liste de « *Reading* » qui inclut les valeurs des *ReadingTypes*.

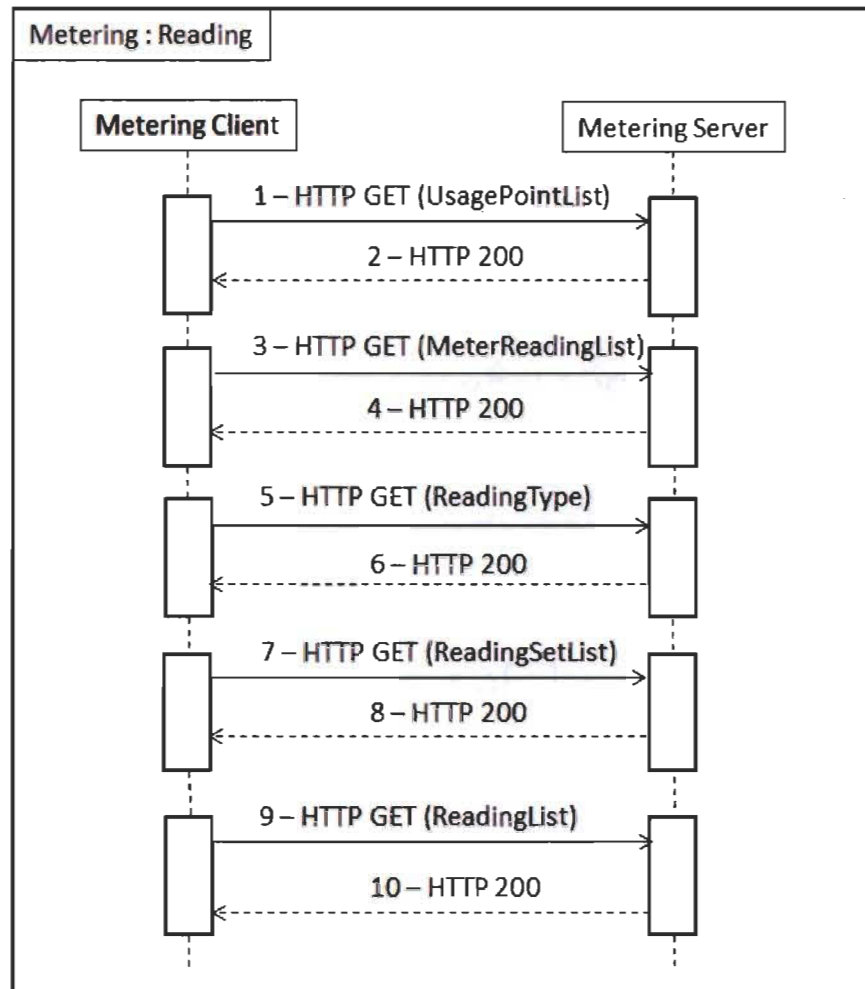


Figure 4.2 Un exemple de la fonctionnalité de mesurage [51]

D'abord, la fonctionnalité de tarification offre multiples types de tarification. Par exemple : la tarification en temps-réel ou la tarification selon l'heure de la consommation. L'information quant au prix soutient plusieurs monnaies, pour les régions internationales. D'ailleurs, les dispositifs intelligents peuvent réagir au signal de prix qui provient de l'utilité. Lorsque le coût est plus élevé que celui prédéfini, les appareils (ex. : thermostat intelligent) agissent pour diminuer la consommation [56].

La Figure 4.3 montre un exemple du processus pour un appareil de client qu'obtient l'information de prix basée sur le temps d'utilisation. Premièrement, le client doit obtenir le

profil de tarif (*TariffProfile*) qui est une structure de tarif, pour cet exemple, le tarif dépend de la structure du temps utilisé. Après avoir reçu la réponse, le client demande une liste de « *RateComponent* ». Ensuite, comme l'exemple le quel présente par la figure 4.2, le client demande « *ReadingType* » dans l'étape 5. Pour l'étape 7 à 10, le client reçoit la liste de la tarification variable dans les différentes périodes de la journée, ainsi que la liste de la tarification qui dépend de la consommation d'énergie.

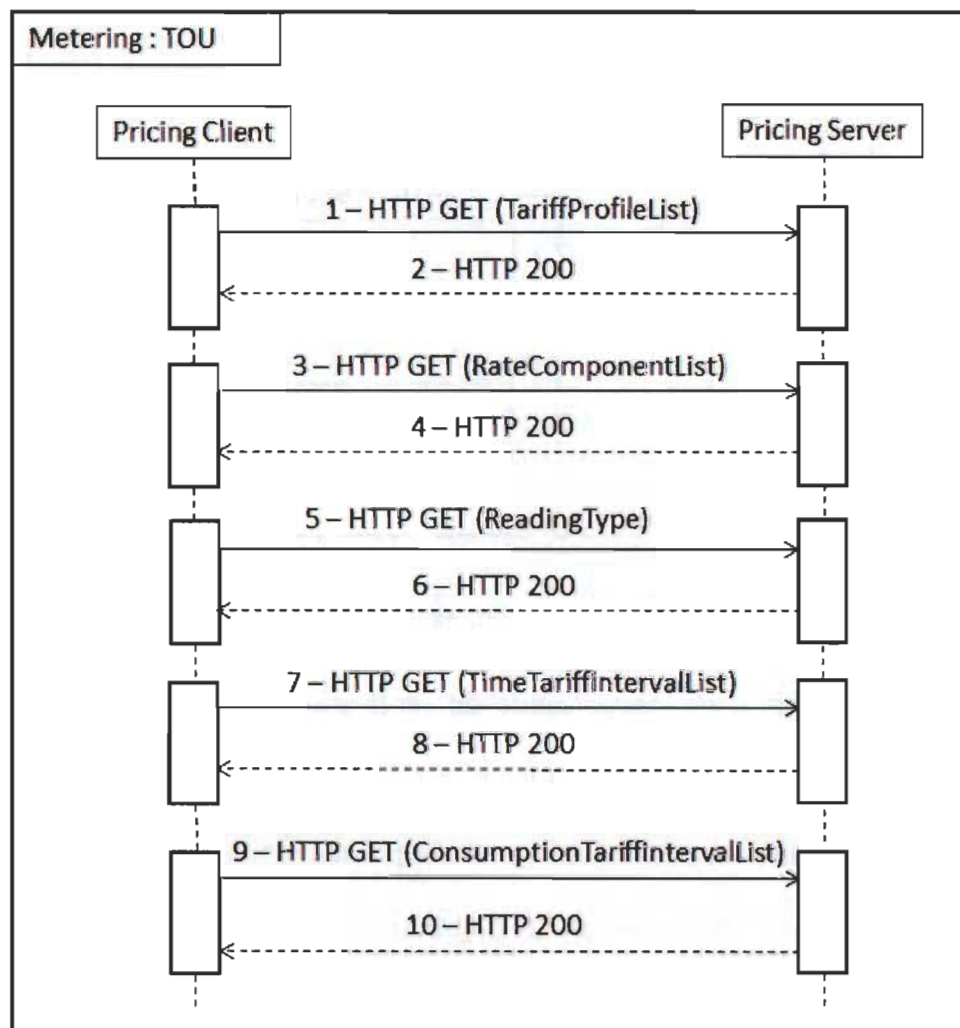


Figure 4.3 Un exemple de la fonctionnalité de tarification [51]

La fonctionnalité de DRLC est ciblée selon les dispositifs du client (ex.: thermostat) et du serveur (ex.: Energy Services Interface). Avec cette fonctionnalité, l'événement de DRLC pourrait être programmé selon les dispositifs spécifiques, destinée à réduire la charge. Plusieurs méthodes peuvent être envisagées pour le contrôle de la charge, y compris la valeur de la température définie et la compensation, le niveau criticité et le cycle de travail. Les dispositifs ont la capacité d'être groupés afin de synchroniser leurs actions. Bref, la fonction de DRLC est utilisée pour éviter le pic de demande et aider les clients à abaisser la dépense d'énergie [51], [54].

Un exemple des processus pour exécuter une fonctionnalité générale de DRLC est présenté sur la figure 4.4. La première étape qui n'a pas été montrée sur la figure inclut les processus lors duquel l'appareil de client cherche les « *function-sets* » dans le serveur, et le ce dernier autorise la connexion. Les deuxième et troisième étapes décrivent la demande de la liste des programmes par l'appareil de client. Plusieurs programmes visent à différents types d'appareil de client, et de la liste « *EndDeviceControl* » qui inclut les événements de contrôle de la charge dans l'appareil de serveur (Energy Services Interface) et les attribues qui servent l'appareil de client à répondre à l'événement lequel de son type. L'appareil de client ensuite répond au côté de serveur (ex : utilité d'électricité) qu'il a reçu l'événement. Le serveur répond à l'appareil de client « 201 » et indique qu'il a créé un nouvel objet. Après, l'appareil de client commence à exécuter l'événement et à envoyer le signal de commencement au serveur. Aussi, lorsque l'exécution est terminée, l'appareil de client va envoyer un signal au serveur.

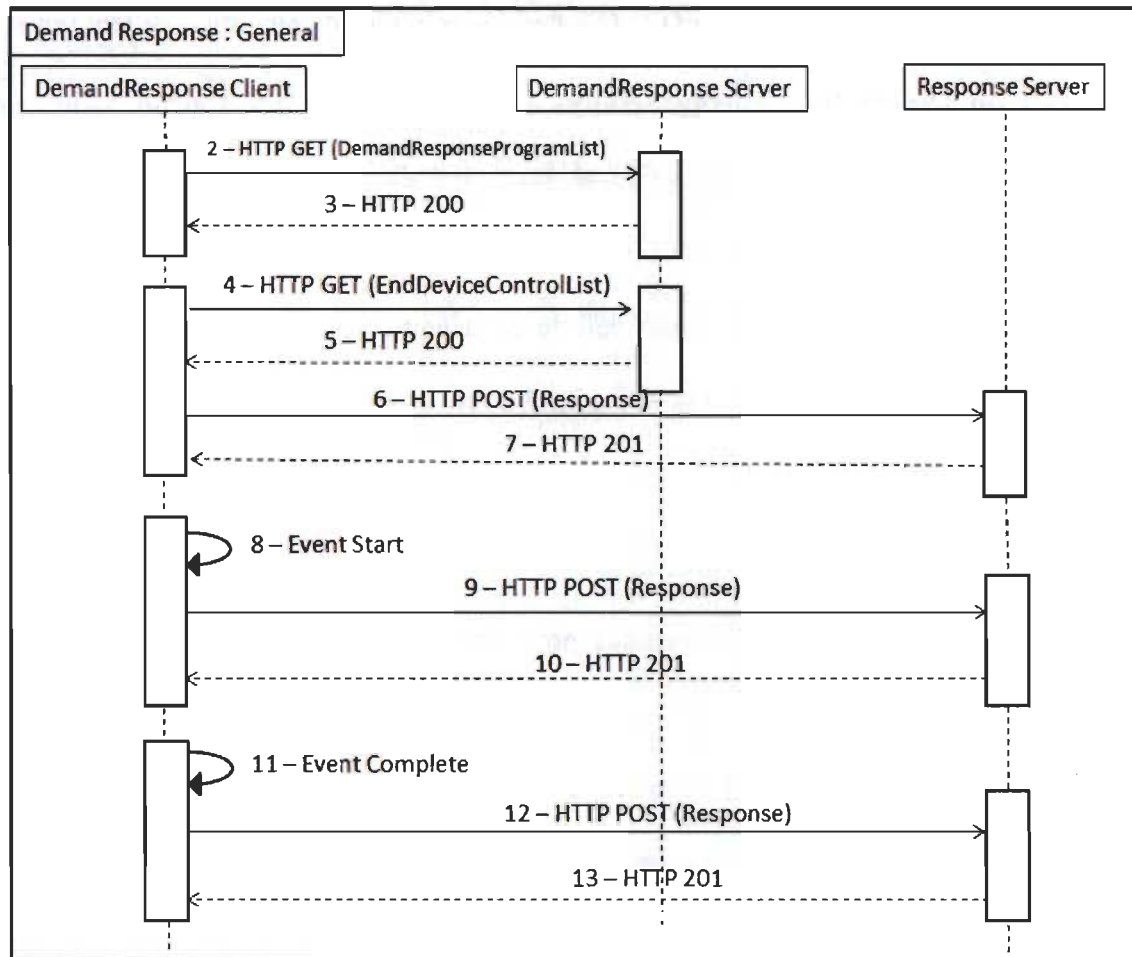


Figure 4.4 Un exemple de la fonctionnalité de DRLC [51]

4.6 Conclusion

Dans ce chapitre nous avons introduit le protocole SEP2.0. Ainsi, le SEP 2.0 est un protocole d'application qui bénéficie à l'interopérabilité de HEMS. Dans la couche physique de SEP2.0, plusieurs technologies de communication sont compatibles telles que HomePlug et ZigBee. Le SEP2.0 permet ainsi la communication par IPv6. Dans SEP2.0, dix « *function-set* » ont été programmées, par exemple la réponse à la demande et le contrôle direct de charges, l'information de compteur et de consommation d'énergie. Ces « *function-set* » aident les développeurs à comprendre la capacité de SEP2.0 et comment il

fonctionne. Le SEP 2.0 emploie l'architecture REST. Dans la couche d'application, il supporte les technologies afin de développer une application web. D'ailleurs, le mécanisme de sécurité est nécessaire pour la couche d'application. Le protocole TLS est utilisé pour assurer la sécurité de SEP2.0. À la fin du chapitre, nous avons montré des exemples concernant les processus des fonctionnalités principales de SEP2.0.

Chapitre 5 - L'application d'interopérabilité de la gestion d'énergie résidentielle

Le concept de réseau intelligent exige normalement la coopération d'utilisateur et d'utilité, afin de réaliser l'objectif d'économiser l'énergie et de fournir un environnement confortable. Toutefois, dans cette étude, on se concentre sur l'application au niveau de la gestion résidentielle (HEMS). Les applications visent à réaliser une maison automatique; contrôler des charges correspondant au besoin personnalisé; fournir une interface pour les opérations d'interopérabilité, etc. Selon la description formulée dans le chapitre 2 et l'architecture de HEMS dans la figure 2-1, un HEMS consiste en appareils d'information, appareils ménagers, et appareils de communication.

À la faveur des nouvelles technologies de communication et passerelle, l'interopérabilité de chaque partie dans une résidence intelligente devient possible. Après la comparaison des technologies de communication pour le réseau local de communication, ZigBee se démarque des autres par sa faible consommation d'énergie et sa popularité dans les produits commerciaux. Nous avons pris pour exemple de construire un réseau local avec des fonctions de surveillance et contrôle de charges qui sont un chauffage et une bouilloire électrique. La passerelle étant un élément essentiel de HEMS, celui-ci s'occupe d'échanger l'information avec les appareils intelligents à distance; ainsi, pour assurer la compatibilité des réseaux qui ont différentes technologies de communication. Dans certains cas, elle offre l'interface pour obtenir les données des appareils connectés, transmettre des messages entre

l'utilité et le client, envoyer les consignes du client à l'appareil indiqué. Il existe de nombreux types de passerelles fabriquées par divers manufacturiers. Par exemple, Netgear produit la passerelle « Home Lifestyle Gateway, model: ASG1100-100NAS » [57]. Cette passerelle offre la connectivité à un système de sécurité, elle est compatible avec les appareils de Wi-Fi et ZigBee et peut fournir une automatisation de la résidence incluant le contrôle de l'éclairage et de charges thermostatiques (chauffage, chauffe-eau).

Bien qu'il existe une grande quantité de choix, ceux-ci ne sont pas tous disponibles pour développer des applications du consommateur. Afin de pouvoir développer une application « propre » (custom application), dans cette étude nous avons considéré les produits de la compagnie Digi International. La compagnie Digi se concentre sur le domaine d'IoT (Internet of Thing), surtout le module de ZigBee qui s'appelle Xbee, et qui permet aux consommateurs de développer des applications spécifiques [58].

Dans ce chapitre, nous présentons les résultats expérimentaux suite à mise en œuvre d'un réseau incluant les passerelles de Digi. Dans un premier temps, nous analysons la construction du réseau de ZigBee pour contrôler la température d'une chambre; dans un deuxième temps nous avons essayé de faire l'implémentation du protocole SEP; et enfin, une interface Web a été développée pour contrôler les appareils dans le réseau de ZigBee.

5.1 Matériel utilisé

Certes, des nombreux produits de ZigBee sont disponibles, par rapport aux autres manufacturiers, les produits de la compagnie Digi se distinguent par la plate-forme de développement ouverte et d'autres avantages pour les développeurs. La compagnie Digi Internationale, telle que mentionnée auparavant, est spécialisée dans la connexion

« machine to machine (M2M) » sans-fil et les applications des Internet des Objets (IoT – Internet of Things). Ses produits dont l'élément principal est le module sans-fil (XBee), qui permettent aux clients de créer des produits, et ce, pour différentes applications spécifiques. Les développeurs mondiaux peuvent profiter des informations techniques, forums de discussion, des plateformes spécifiques pour développer les applications, et l'accès au « stockage dans les nuages » et à des applications Web pour le contrôle et la surveillance à distance.

Les produits de Digi comprennent des modules RF, adaptateurs, capteurs, passerelles, etc. Dans nos expérimentations, nous avons employé les prises intelligentes (smart plug), le capteur de température, ainsi que la passerelle ZigBee et la passerelle ZigBee-SEP.



Figure 5.1 Produits de Digi utilisés dans les expérimentations

La prise de courant intelligente de Digi, muni d'un module RF XBee, a la capacité de mesurer et de contrôler les dispositifs branchés pour atteindre l'objectif d'utiliser l'énergie efficacement et de diminuer la dépense. Les données qu'elle mesure incluent la température, le courant et la lumière [59]. Le capteur ZigBee détecte également la température, la radiance, l'humidité [60]. L'utilisation de la passerelle XBee centralise plusieurs appareils XBee dans un HEMS. D'ailleurs, la passerelle permet la connexion de WiFi, ZigBee, et Ethernet. Le réseau local, peut être configuré et géré à distance via la solution

inonuagique de Digi [61]. Avec un environnement de développement ouvert sous Python, les clients sont capables d'exécuter leurs propres applications dans la passerelle.

5.2 Construction d'un réseau ZigBee pour contrôler la température d'une chambre

L'expérimentation vise à construire un petit réseau local pour analyser la performance et la fonctionnalité des dispositifs ZigBee. Ce réseau est muni d'une passerelle de XBee, une prise intelligente, un chauffage portatif, et un ordinateur. La structure du réseau est montrée dans la figure 5-2. La passerelle communique avec une prise intelligente via le signal de ZigBee, et avec un ordinateur par Ethernet. Le chauffage portable branché sur la prise intelligente et permet de contrôler la température de la pièce en modulant la puissance de chauffage. Le but de l'application devrait être de contrôler automatiquement le chauffage, afin de garder la température de la pièce près de celle configurée dans le programme.

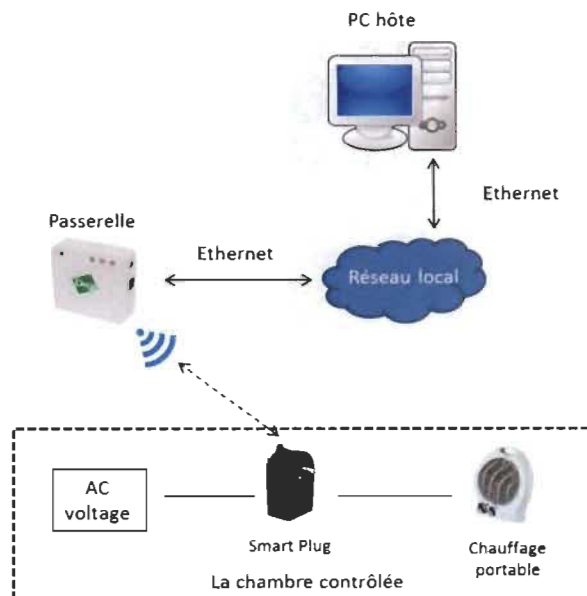


Figure 5.2 Structure d'un réseau de ZigBee pour le contrôle de température

5.2.1 L'environnement de programmation -- ESP

Tel que mentionné dans les sections précédentes, les produits de Digi supportent le langage python pour le développement des applications. Python est un langage de programmation orienté objet, parmi ses avantages on peut mentionner sa simplicité, et la disponibilité de plusieurs bibliothèques logicielles. Il est alors possible de réaliser une application en peu de lignes de programme. L'outil proposé pour la programmation des applications incluant des appareils Digi en Python s'appelle ESP. L'ESP est un puissant IDE (Integrated Development Environment), qui fournit aux développeurs une plateforme de développement adaptée pour les produits Digi. Il s'agit d'un éditeur de code source, débogage, interpréteur python, gestion de projet, service de terminal distant, etc. La figure 5-3 présente la structure d'ESP [62].

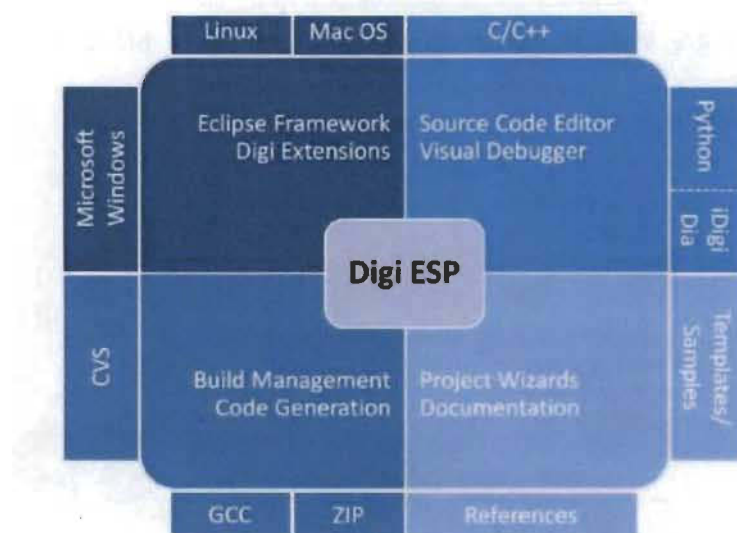


Figure 5.3 Structure de Digi ESP [62]

Après l'installation du matériel, sur l'interface de gestion d'appareil dans ESP, la passerelle peut être trouvée par deux moyens. Le premier est par la connexion de LAN, car la passerelle et l'ordinateur existent dans la même LAN. L'autre moyen de trouver la

passerelle est par la connexion infonuagique, si le réseau local a une connexion Internet. L'utilisation du service infonuagique de Digi (Digi-nuages) peut poser certains inconvénients si la condition de l'accès d'Internet pourrait ne pas être satisfaite en tout temps, et aussi car le service entraînera peut-être des frais supplémentaire. Dans ce travail nous avons opté pour la connexion de réseau local (LAN) pour la configuration de la passerelle.

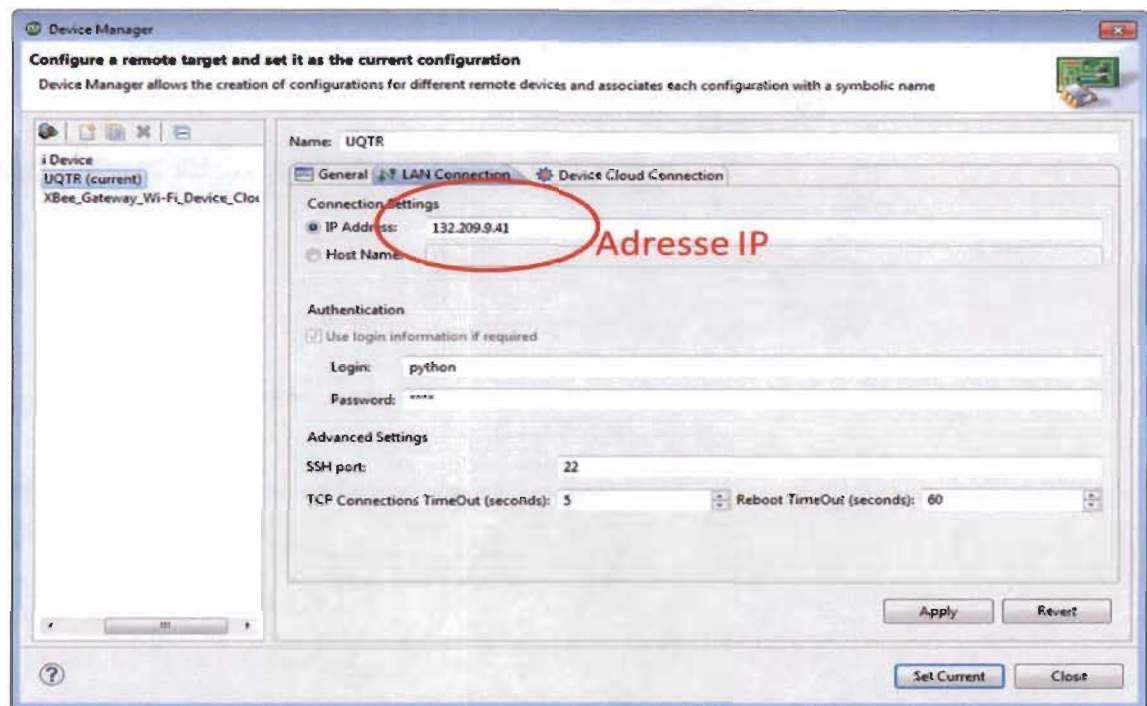


Figure 5.4 Interface de la gestion de dispositif

5.2.2 L'interface web locale pour la configuration des dispositifs - XBee

La passerelle est le centre et le coordonnateur du réseau ZigBee. Avec l'adresse IP de la passerelle, une interface locale est disponible afin de faire la configuration du réseau. La Figure 5-5 montre la page principale de la gestion et montre la configuration des appareils dans ce réseau ZigBee. Sur la page principale, les informations de la passerelle et les états de connectivité du réseau local et d'Internet sont présentés. L'information des appareils

connectés à la passerelle doit être fournie dans la partie de « XBee Network » qui se trouve sous le menu de configuration. Dans ce cas, il y a seulement une prise intelligente connectée avec la passerelle, qu'illustré sur la figure 5-6. Dans la page de « XBee Network », on peut trouver aussi les adresses étendues des appareils. Cette adresse est unique pour chaque dispositif, elle constitue une clé pour échanger l'information à la destination exacte.

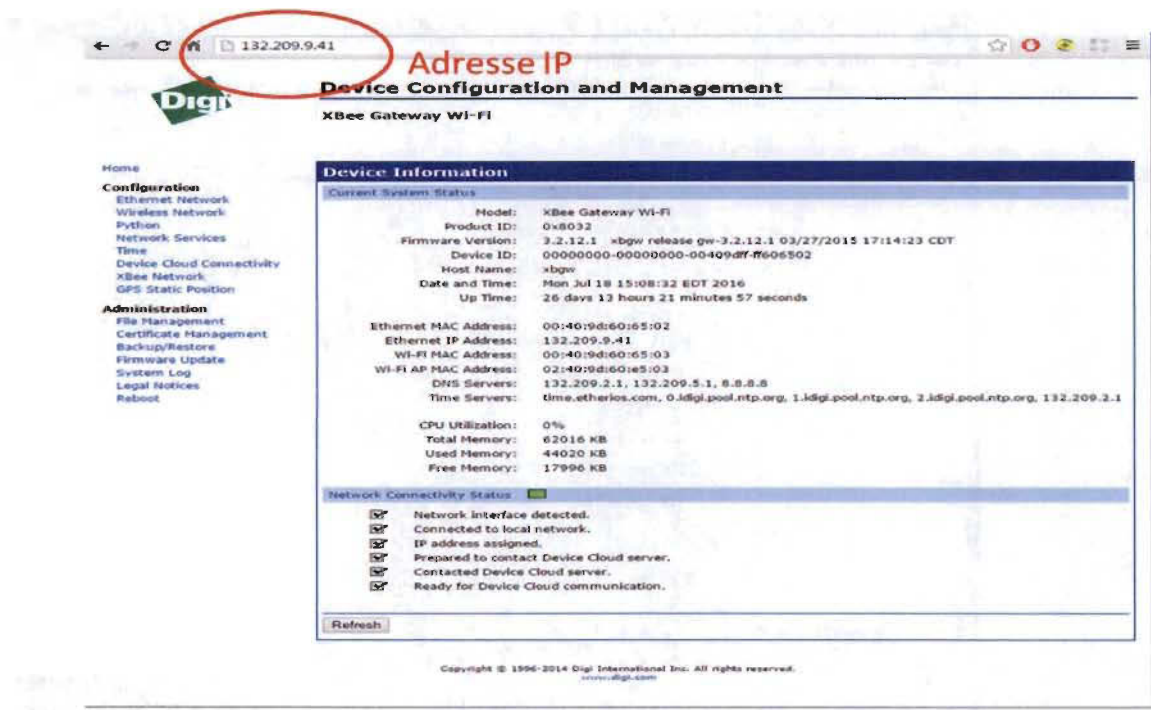


Figure 5.5 Interface de web local pour la gestion de dispositif

XBee Configuration

▼ XBee Devices

XBee Device on the Gateway

Network PAN ID: 0x6848 - 0x0000000000001234
 Network Channel: 0x0f (2425 MHz)
 Gateway XBee Address: 00:13:a2:00:40:b9:cf:64
 Gateway XBee Firmware: 0x4043

Select the Gateway XBee device to configure it:

| Node ID | Network Address | Extended Address | Node Type | Product Type |
|--------------|-----------------|-------------------------|-------------|--------------|
| XBee Gateway | [0000] | 00:13:a2:00:40:b9:cf:64 | coordinator | XBee Gateway |

Remote XBee Devices

Select a device to configure:

| Node ID ▲ | Network Address | Extended Address | Node Type | Product Type |
|------------|-----------------|-------------------------|-----------|--------------|
| smart plug | [0708] | 00:13:a2:00:40:6e:3f:49 | router | Smart Plug |

1 router

Discover XBee Devices

☒ Clear list before discovery

▶ OTA Firmware Update Setup

▶ OTA Firmware Update

Copyright © 1996-2014 Digi International Inc. All rights reserved.
 www.digi.com

Figure 5.6 Page de configuration de XBee

La prise intelligente de XBee utilise les commandes AT pour la consultation, la configuration et le contrôle du module RF XBee à l'intérieur de la prise. Le numéro de la commande peut être utilisé pour recueillir les données numériques et analogiques dans la partie de command I/O. Ainsi, la commande D1 est destinée à l'information de capteur de radiance; D2 est destinée au capteur de courant; et D4 est utilisée pour contrôler l'état de conduction (fermée/ouverte) de la prise. La lecture des capteurs est habilitée lorsque les commandes D1 et D2 sont fixées à 2. Pour le contrôle de la prise, si la valeur d'AT D4 est à 4 (état « Low »), la prise est ouverte et il n'y a pas de courant. Si elle est à 5 (état « High »), la prise est fermée et la charge connectée à la prise est alimentée.

Input/Output Settings

I/O Pin Settings

| DIO | AT | Functions | Setting | Pull up | Detect |
|-----|----|---------------|--|-------------------------------------|-------------------------------------|
| 0 | D0 | AD0, CB | Commissioning Button (1) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 1 | D1 | AD1 | Analog Input (2) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 2 | D2 | AD2 | Analog Input (2) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 3 | D3 | AD3 | Analog Input (2) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 4 | D4 | | Digital Output, High (5) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 5 | D5 | Associate LED | Disabled (0) Digital Input (3) Digital Output, Low (4) Digital Output, High (5) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 6 | D6 | RTS | Digital Output, Low (4) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 7 | D7 | CTS, RS-485 | CTS Flow Control (1) | <input type="checkbox"/> | <input type="checkbox"/> |
| 10 | P0 | RSSI PWM | RSSI PWM Output (1) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 11 | P1 | | Disabled (0) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| 12 | P2 | | Disabled (0) | <input checked="" type="checkbox"/> | <input type="checkbox"/> |

Pull-up Resistor Enable (PR): bitfield (0-0x7fff)

DIO Change Detect (IC): bitfield (0-0xffff)

I/O Settings

Associate LED Blink Time (LT): x 10 msec (10-255, 0=default)

RSSI PWM Timer (RP): x 100 msec (0-255)

I/O Sampling Rate (IR): msec (32-65535, 0=disabled)

Supply Voltage High Threshold (V+): mvolts (0-65535)

Figure 5.7 Interface Web pour configurer les entrées et sorties (I/O)

5.2.3 La programmation de l'application

Spécifiquement, le programme d'application sur la passerelle devrait accomplir les fonctions suivants : 1) la passerelle cherche et établit la connexion avec la prise intelligente; 2) les données reçues sont correctement interprétées; 3) si la température détectée par la prise est inférieure à une valeur de référence, la prise permet l'alimentation de la charge; au contraire, si la température dépasse la valeur de référence la prise doit couper l'alimentation de la charge.

Le développement de l'application profite d'ESP pour éditer, déboguer et interpréter le code source. En vue d'accomplir la constitution de la connexion entre la passerelle et la prise, une fonction nommée `socket()`, ainsi qu'un objet de `socket` – `socket.bind()` sont employés. Voici un exemple de `socket`:

```
s = socket(AF_XBEE, SOCK_DGRAM, XBS_PROT_TRANSPORT)
```

```
s.bind(("", 0xe8, 0xc105, 0x92))
```

Les données reçues sont toujours selon le format d'API (Application Programming Interface). En se référant à la configuration de la prise, on constate que ses données correspondent au masque analogique qui est « bit field ». Tous les modules XBees ont un convertisseur analogique-numérique (CAN, ou en anglais ADC) de 10-bit, donc un module XBee accepte les données de ADC de 0 à 1023. D'ailleurs, XBee de ZigBee utilise 1.2V tension d'entrée des ADC. Pour les échantillons d'analogiques, les valeurs de 2-bits seront reçues au format hexadécimal. La formule (1) est employée pour transformer les valeurs reçues aux données de tension pratique, où VREF dans l'équation se réfère au voltage de référence qui est 1200 (1,2V) [63].

$$\text{ADC}/1023 (\text{VREF}) = \text{Voltage (1)}$$

Les valeurs d'ADC1, ADC2, et ADC3 de la prise intelligente sont les données pour calculer l'éclairement lumineux, la température et le courant utilisant les équations suivantes [63].

$$\text{Lumière: Lux} = \left(\frac{\text{ADC1}}{1023} \right) * 1200 ;$$

$$\text{Température: Temp}_c = \frac{\left(\left(\frac{\text{ADC2}}{1023} \right) * 1200 \right) - 500}{10} ;$$

$$\text{Courant: Courant} = \frac{\left(\left(\frac{\text{ADC3}}{1023} \right) * 1200 \right) * \left(\frac{156}{47} \right) - 520}{180} * 0.7071$$

La passerelle met à jour les informations à une période prédéfinie. Dans le cas spécifique des tests réalisés, lorsque la valeur de la température est supérieure à 25°C, une commande est envoyée pour changer la valeur de D4 de la prise à 4 afin de couper le courant de sortie; conséquemment, le chauffage arrête de fonctionner. Une fonction `ddo_set_param()` est utilisée pour réaliser la transmission de la commande. Par exemple, l'adresse de cette prise est [00:13:a2:00:40:6e:3f:49], alors la ligne de commande complète doit être : `ddo_set_param('[00:13:a2:00:40:6e:3f:49]!', 'D4', 4)`.

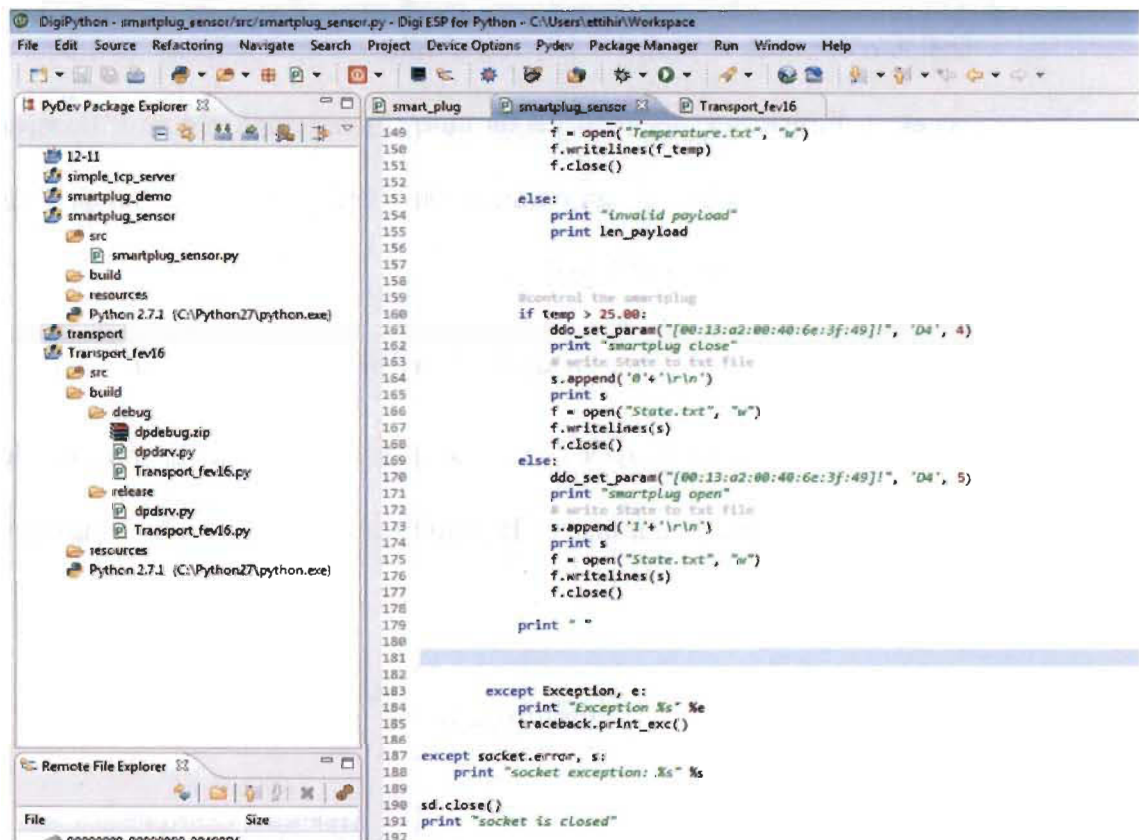


Figure 5.8 Aperçu de l'interface de la plateforme de développement Digi-ESP

Après l'accomplissement du programme, il y a deux façons d'effectuer la mise en place. La première consiste à l'exécuter sur la plateforme ESP. Cette façon est plus pratique, car le processus de débogage est disponible et l'interpréteur visible. Par contre, il faut

toujours ouvrir ESP et exécuter le débogage et l'interpréteur du programme. L'autre façon consiste à enregistrer le programme de Python, dans la passerelle par l'interface Web locale. Lorsque le programme est autorisé par la passerelle, il s'exécute automatiquement.

Compte tenu du côté pratique de l'application, nous avons enregistré le programme, qui a déjà été débogué et interprété par ESP à la passerelle (figure 5-9), et activé sur la page de configuration de Python (figure 5-10).

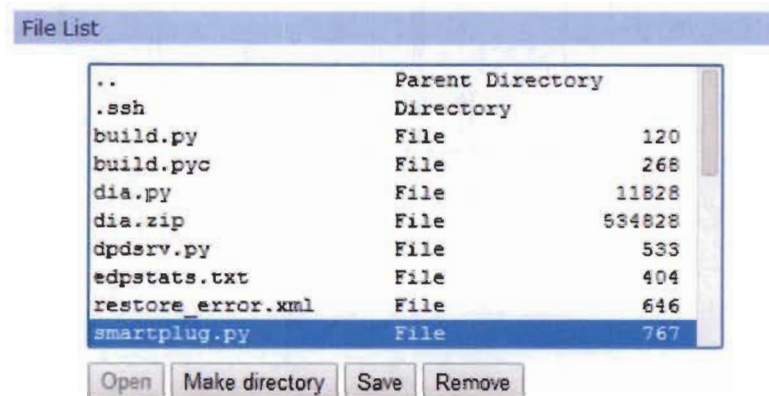


Figure 5.9 Liste des fichiers enregistrés dans la passerelle

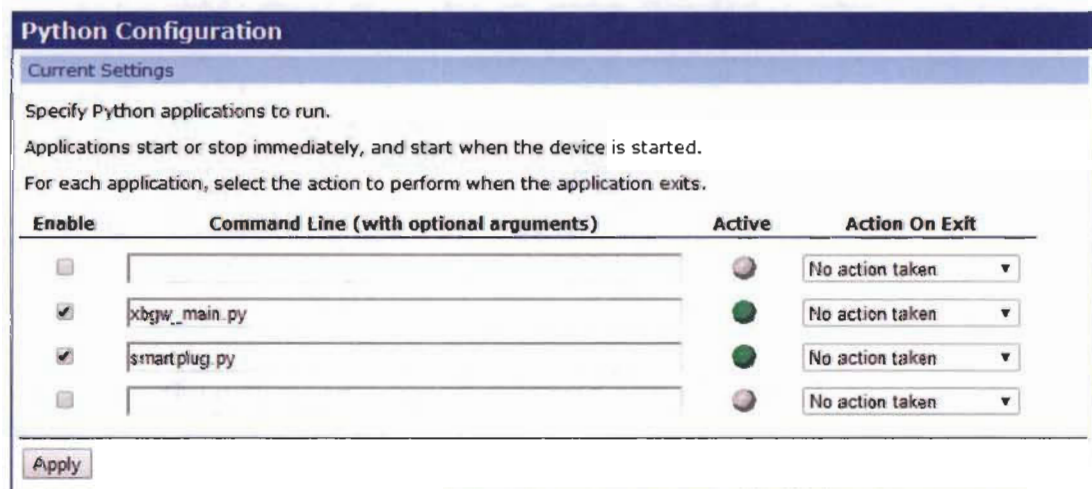
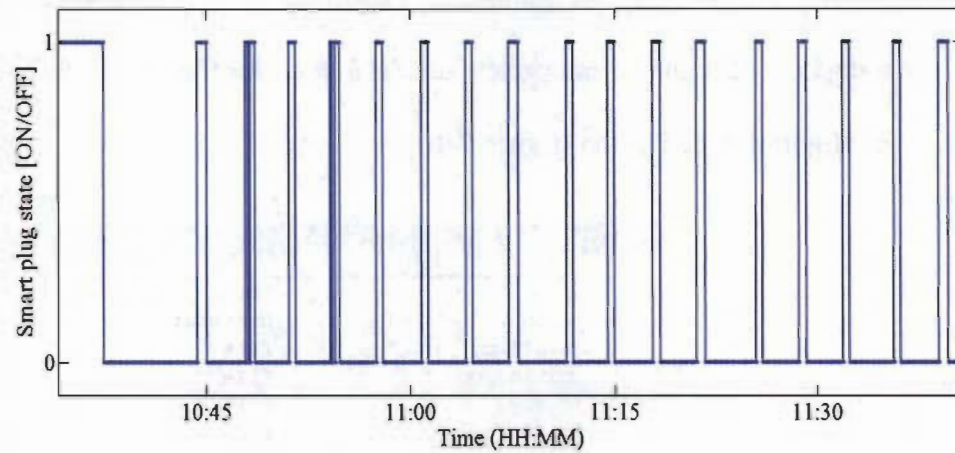
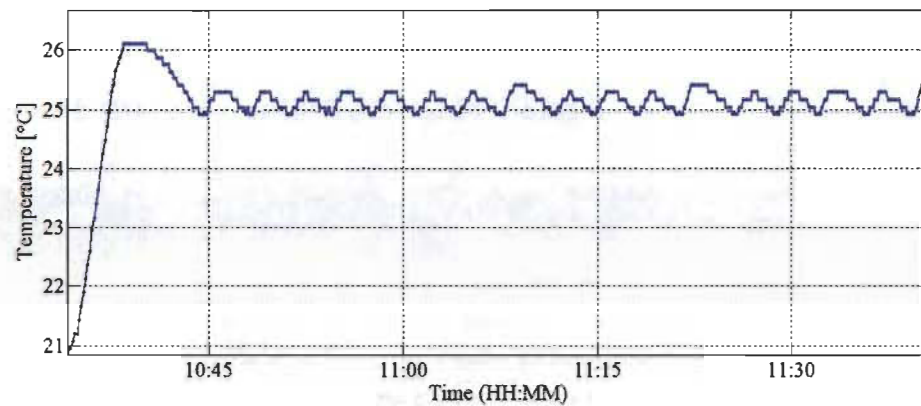


Figure 5.10 Activer le programme

Afin de faciliter l'analyse, les résultats et les données sont enregistrés dans des fichiers de texte (*.txt). La figure 5-11 présente le plot des résultats de l'expérimentation qui contient les températures mesurées et les états de la prise pendant une heure d'opération.



(a)



(b)

Figure 5.11 Résultats d'expérimentation a) l'état de la prise intelligente (1:ON, 0:OFF) et b) la température de la chambre

Les résultats démontrent que la prise intelligente fonctionne exactement comme l'objectif du programme d'application implanté sur la passerelle XBee. Il est important de

noter que dans le test, le débit de données est défini à 10 secondes. C'est-à-dire que la passerelle interroge pour une nouvelle donnée et met à jour l'état de la prise toutes les 10 secondes. Il faut remarquer que la température de la chambre est bien contrôlée et, lorsque le système est stabilisé, l'erreur absolue moyenne de température (en régime établi) reste inférieure à 0,5°C.

5.3 Construction d'un réseau ZigBee-SE

Comme l'énergie intelligente est la tendance du réseau intelligent et qu'il bénéficie de la gestion et de l'économie d'énergie, cette expérimentation vise à construire un réseau de ZigBee SE avec une passerelle SE de Digi. La passerelle d'énergie intelligente équipée de la certification de SEP-1, permet les applications de gestion de la demande et contrôle direct de charges, d'efficacité énergétique et d'engagement du consommateur [64]. Tous les dispositifs intégrés à un réseau de ZigBee-SE doivent être installés selon la certification SE. Si le dispositif n'a pas cette certification, il ne peut pas se connecter avec la passerelle en toute sécurité. Les développeurs peuvent demander un certificat de test sur le site Web de Certicom [65].

Théoriquement, un module de ZigBee doit pouvoir être converti au ZigBee-SE module, simplement par l'installation d'une certification, et par la mise à jour du firmware au SE.

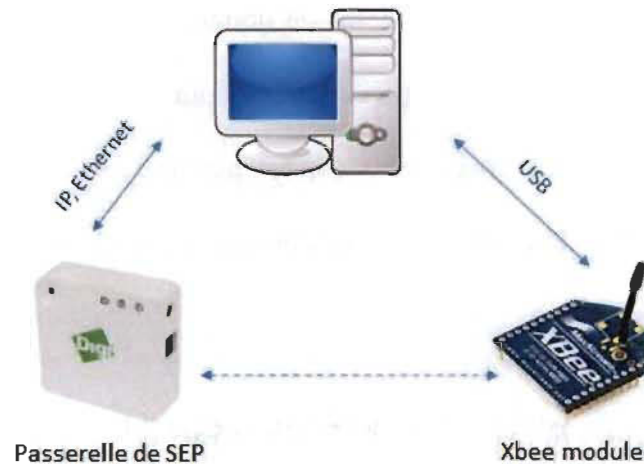


Figure 5.12 Réseau de SEP

Dans le cadre du projet, nous avons essayé la conversion de modules ZigBee à ZigBee-SE; toutefois, malgré avoir réussi à installer la certification et réalisé la mise à jour du firmware nous n'avons pas réussi à faire fonctionner correctement le module converti à ZigBee-SE. Il faut souligner, que l'utilisation de la passerelle de SEP est vraiment utile lorsqu'on prévoit une gestion à distance notamment à partir de l'utilité, dans ce cas utilisant des produits qui sont à l'origine certifiées ZigBee-SE.

5.4 Développement d'une application pour la gestion locale avec ZigBee

Le contrôle des charges dans un réseau local, tel qu'il a été présenté auparavant, est possible par l'emploi de la passerelle et des dispositifs intelligents. En complément, il pourrait être très utile de posséder une application Web pour permettre au client de surveiller l'état du système et pour implanter, le cas échéant, des stratégies de gestion automatisées. Dans une telle l'interface Web, les états des opérations des charges sont visibles et les commandes d'utilisateur sont pratiques à envoyer. Dans cette section, les

étapes principales et les technologies utilisées pour développer l'application Web basée sur un réseau local et une passerelle ZigBee, sont présentées.

5.4.1 Structure du réseau pour l'application

La structure du réseau de cette expérimentation est présentée dans la figure 5-13. Ce réseau permet d'intégrer un capteur qui détecte la température, l'éclairement lumineux et l'humidité; ainsi deux prises intelligentes alimentant une bouilloire électrique et un chauffage d'appoint respectivement. Le site Web doit alors permettre l'affichage des données du capteur et des prises intelligentes; ainsi que l'envoi de signaux de commande pour contrôler l'alimentation des charges.

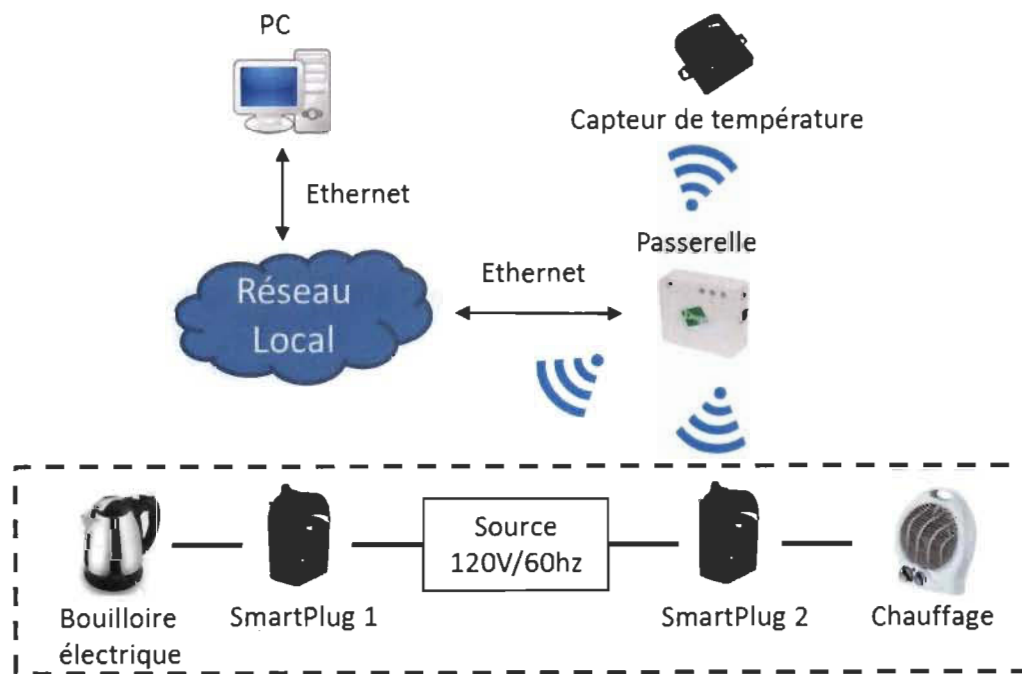


Figure 5.13 Structure d'application Web de HEMS

5.4.2 *Processus de développement de l'application Web*

Une application permettant de contrôler une résidence intelligente consiste en au moins deux grandes parties : un programme qui s'exécute dans la passerelle et un programme pour l'interface Web; la relation entre eux est de type serveur - client.

Le programme exécuté dans la passerelle permet la transmission de l'information ou de commandes entre les dispositifs intelligents et l'ordinateur hôte. Les fonctions principales de ce programme sont l'obtention des informations des dispositifs intelligents; détection l'adresse IP des clients qui essaient de se connecter à la passerelle; transmission de l'information au client et à envoyer la commande du client aux dispositifs de contrôle (prise intelligente). Certaines technologies et certains outils permettent de réaliser cette partie, comme ESP et socket, qui sont abordés dans la section 5.2. Il est à noter que le lien entre la passerelle et l'ordinateur hôte est l'adresse IP et non pas l'adresse MAC.

À part le programme principal, un autre programme subalterne est enregistré dans la passerelle. Ce programme subalterne génère et renouvelle une liste d'adresses des dispositifs ZigBee connectés, car le nombre de dispositifs est incertain dans les cas réels. Le programme principal de l'application acquiert les informations des dispositifs dans la liste, et la passerelle surveille toujours les connexions des clients. Évidemment, il est probable que plus d'un client essaie de se connecter à la passerelle en même temps; ainsi, afin de gagner du temps et de l'espace, les processus doivent être asynchrones. Pour la programmation en Python, la fonction « select » du module « select » est disponible pour réaliser les communications asynchrones. La structure de « select » est : `select.select (rlist, wlist, xlist[, timeout])` [66]. Les trois premiers paramètres sont essentiels, ils repensent respectivement les listes de l'objet à lire, l'objet à écrire et l'objet avec exception. Le

dernier paramètre réfère le temps d'attente [66]. La passerelle doit lire les données des dispositifs XBee et les commandes des clients; si un dispositif XBee reçoit une commande, il doit être mis dans la liste « wlist » et lorsqu'un client est connecté, il est écrit avec les informations de dispositifs XBee. Par conséquent, les deux listes « rlist » et « wlist » incluent toutes deux la liste du dispositif XBee et la liste des clients connectés par TCP. La figure 5-14 décrit généralement l'organigramme de programmation de cette partie qui s'exécute sur la passerelle.

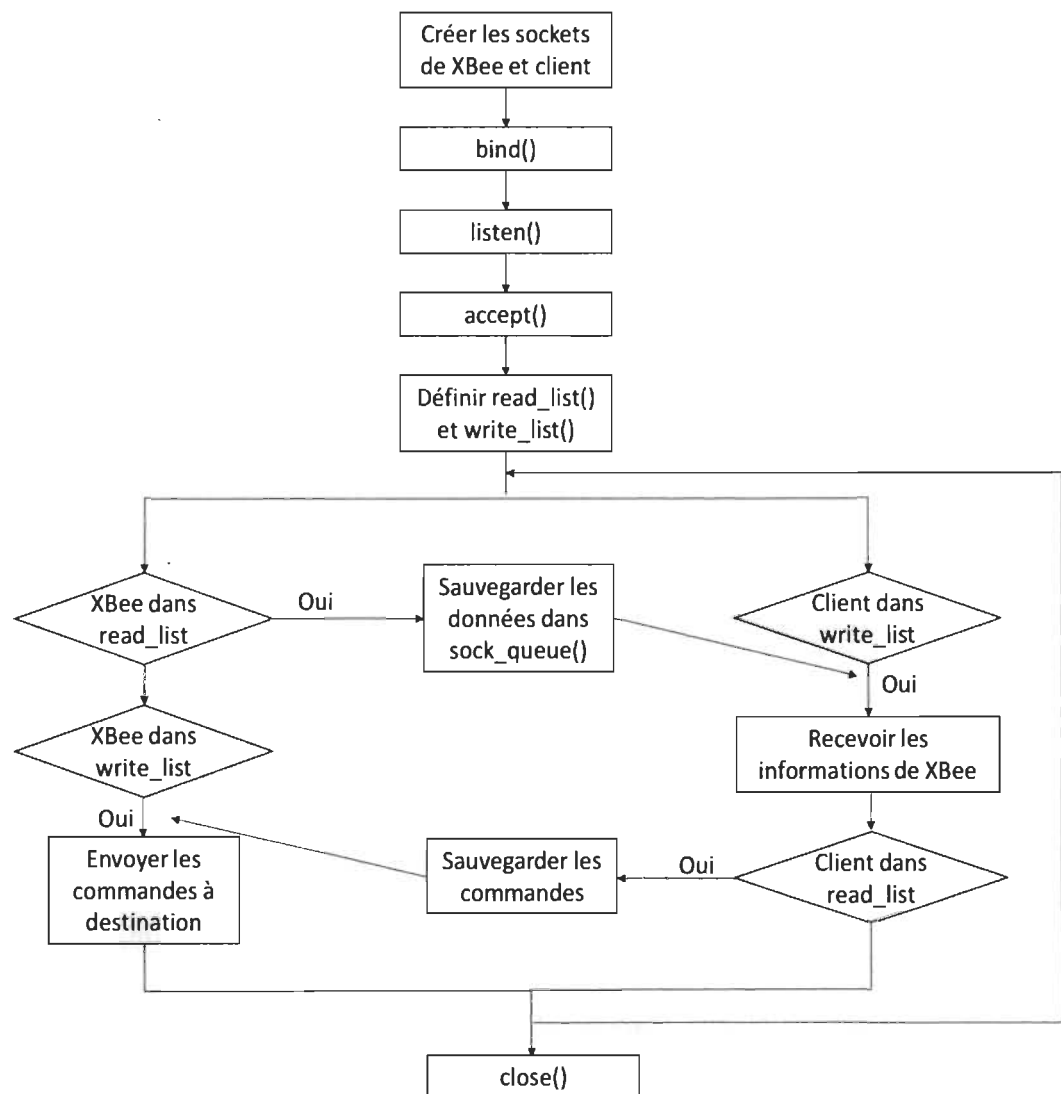


Figure 5.14 Organigramme de programmation

La programmation qui accomplit l'interface de page Web contient deux parties. Une première partie permet de configurer la représentation ou l'apparence de la page Web; et l'autre permet de rendre la page Web dynamique afin de mettre à jour les informations affichées en fonction des informations de l'état actuel du système reçues en temps-réel.

Le HTML (HyperText Markup Language) est utilisé pour le développement de la représentation de la page Web, est un langage de balisage, qui offre un standard pour la description de la page Web. Le navigateur exécute le document en format HTML et représente la page Web en fonction des balisages marqués. Une page HTML contient au moins les trois éléments suivants :<html></html>, <head></head> et <body></body>. La balise <html> se situe au début de document, afin de déclarer que ce document est écrit en HTML. Entre les balises <head> et </head>, les textes comportent les informations sur la page web, par exemple le titre de page, la préface et la démonstration, etc. qui ne sont pas affichés dans la page. De plus, les contenus exacts présentés sur la page sont incorporés dans les balises <body> et </body>. En conséquence, la plus simple structure d'un HTML peut être organisée comme la figure 5-15.

```
<html>
  <head>
    <!-- l'information de la page web -->
  </head>
  <body>
    <!-- les contenus de la page web -->
  </body>
</html>
```

Figure 5.15 La structure simple de HTML

Idéalement, une page Web devrait être développée en combinant le HTML avec les autres technologies du Web, par exemple le langage de script, l'interface de passerelle commune, Framework. Dans l'application développée dans le cadre du projet, HTML

fonctionne avec CSS (Cascading Style Sheets) et JavaScript. Dans ce cas, le HTML décide du contenu présent sur la page Web; CSS s'occupe de changer l'apparence du contenu tel que le style de caractère, le fond d'écran, la bordure; et puis JavaScript, qui est un langage de programmation de script, permet l'exécution des fonctions et actions intégrées dans la page Web [67].

À l'aide de CSS, la position de chaque élément peut être contrôlée précisément, comme l'efficacité de développement est augmentée[68]. Dans notre page HTML, nous employons CSS avec la balise `<div>` `</div>` pour concevoir la disposition de la page Web. Ce moyen DIV+CSS sépare le contenu de la page et son apparence, simplifie les codes de développement, et augmente en même temps la vitesse d'accès à la page. La principale fonction de JavaScript, dans notre cas, est le changement de la couleur (rouge/vert) du bouton (carré) qui sert à indiquer l'état (de conduction) des prises intelligentes. Si la prise charge est alimentée, JavaScript agit et permet de signaler cette condition en imposant la couleur verte dans la position désignée de la charge en question; en revanche, une couleur rouge est utilisée pour signaler que la charge n'est pas alimentée.

Le langage Python soutient plusieurs « Frameworks » pour le développement Web. Framework offre une structure de développement, comme un produit semi-fini, et sépare l'interface Web et l'application en arrière. Conséquemment, le Framework aide à réduire le cycle de développement et à simplifier la modification de la programmation. Peu importe la modification apportée à l'interface Web ou à la partie d'application, l'autre ne va pas être affectée. Les nombreux Frameworks de python, tels que web.py, Flask et Tornado, ont différentes caractéristiques, ils s'adoptent parfaitement à des situations variées.

Django, comme outil de développement, possède une documentation officielle précise, détient une grande partie du marché, est équipé de fonction sur base de données (MySQL, Oracle, Postgresql...). Django opère en fonction de l'architecture logicielle MVC (modèle-vue-contrôleur), mais se concentre spécifiquement sur MTV (modèle-gabarit/template-vue). La couche modèle s'occupe de traiter les données; la couche gabarit décide de la manière de présenter les données; la couche vue est responsable de l'apparence de la page Web et relie le modèle au gabarit [69]. La relation des trois couches et le processus de Django sont présentés dans la figure 5-16. Étant donné que notre application vise un réseau ZigBee qui a seulement trois dispositifs connectés, l'interface Web est simple, aucun Framework n'est nécessaire. Si cette application peut mettre en place un scénario qui impliquant un grand réseau, la page Web devra accomplir des fonctions complexes et Django pourrait être envisagé dans ce cas pour le développement Web.

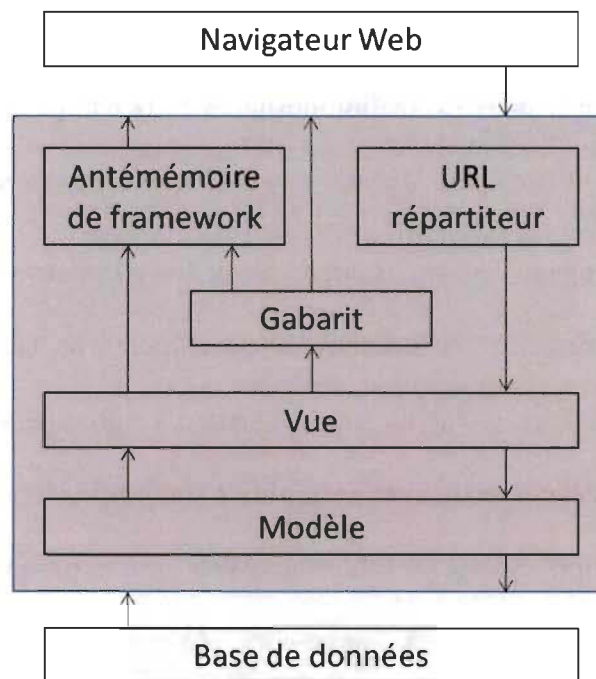


Figure 5.16 Relation de MTV

Nécessairement, une page Web complète doit disposer d'un serveur pour exécuter l'application et répondre aux demandes. Comme indiqué ci-dessus, le développement Web en Python soutient plusieurs frameworks qui sont compatibles avec différents serveurs. Il existe divers outils qui peuvent être intégrés à Python et au serveur, par exemple : CGI (Common Gateway Interface) et FastCGI. Dans notre programmation, nous avons utilisé WSGI (Web Server Gateway Interface) qui est une interface standard entre le serveur Web et l'application Web[70]. Par rapport à CGI, WSGI est conçu originalement pour le langage Python, il sert de pont pour rattacher le HTML et l'application en Python; l'application peut fonctionner sur n'importe quelle interface de passerelle et de serveur.

WSGI est divisé en trois parties : le côté du serveur WSGI, le WSGI Middleware, et le côté de l'application WSGI. Lors du traitement d'une demande WSGI, le côté du serveur doit : 1) d'abord recevoir une demande pour encapsuler les informations d'environnement; 2) ensuite appeler le programme d'application; 3) répondre à l'application avec les informations d'environnement et la fonction de rappel; et 4) après que la partie d'application ait traité les données, il retourne la tête, le corps et l'état au serveur. Middleware est un élément qui n'est pas indispensable. Il est situé entre le serveur et l'application, et il fonctionne sur les deux côtés. Une application WSGI peut être développée par des codes simples, la figure 5-17 en propose un exemple. La fonction `application()` dans l'exemple est une fonction qui se conforme au standard WSGI qui accepte deux paramètres : « environ » et « start_response ». Le paramètre « environ » contient un objet dictionnaire qui inclut les variables d'environnement, comme l'information du serveur ou du client. Ainsi, « start_response » est une fonction qui envoie la réponse de HTTP et sert à transformer la réponse tête/état/corps au serveur WSGI. Dans

notre programmation, la partie du serveur a été réalisée par le module « wsgiref.simple_server ». La figure 5-18 présente un serveur qui appelle la fonction application() dans la figure 5-17. Pour exécuter un serveur, selon l'exemple, nous avons besoin de l'hôte, la porte, et l'application appelée. Lorsqu'un client demande la connexion, le serveur analyse l'information du client et l'insère dans la variable d'environnement « environ », ainsi que l'information du serveur. L'application est appelée par un module « wsgi handler », qui renvoie « environ » et la fonction de rappel à l'application. Ensuite, l'application répond selon la tête/état/corps à « wsgi handler ». Finalement, les réponses sont transformées au client.

```
def application(environ, start_response):
    start_response('200 OK', [('Content-Type', 'text/html')])
    return '<h1>Hello, web!</h1>'
```

Figure 5.17 Un exemple de WSGI application

```
from wsgiref.simple_server import make_server, application

httpd = make_server('', 8000, application)
print("Serving HTTP on port 8000...")

httpd.serve_forever()
```

Figure 5.18 Un exemple de serveur WSGI

En combinant la théorie avec notre application, le processus de réalisation de l'interface Web se lirait de la façon suivante :

- 1) L'application sur la passerelle ouvre les sockets pour les dispositifs XBee et le client (pour la passerelle, l'application de WSGI agit comme un client). La passerelle reçoit les données des dispositifs XBee et recueille la connexion du client;

2) Exécute le code de HTML via le serveur dans l'ordinateur hôte. S'il y a des clients qui accèdent à la page Web, le serveur appelle l'application de WSGI et envoie les informations des clients du côté de la passerelle. La passerelle transforme tout de suite les données des dispositifs XBee au serveur;

3) L'application de WSGI tarit les données reçues et répond au serveur avec la tête de la page Web, l'état de la page, et le corps qui sont présents sur la page;

4) Si les clients demandent à changer les états des prises intelligentes, les commandes seront reçues par l'application de WSGI et transformées à la passerelle par socket;

5) Après que la passerelle ait reçu les commandes, elle va les envoyer au dispositif destiné par socket.

5.4.3 Le résultat de l'expérimentation du développement d'une application web

Afin d'accomplir l'application Web, il faut tout d'abord procéder à l'installation du matériel afin d'établir le réseau. Ensuite, il faut sauvegarder le programme qui gère les dispositifs XBee et les données dans le réseau XBee vers la passerelle. Finalement, il s'agit d'exécuter le programme qui comprend le HTML et l'application WSGI sur l'ordinateur hôte. Après toutes ces étapes, mettre l'adresse du serveur avec la porte définie dans un navigateur; l'interface se présentera comme dans la figure 5-19. Ici dans l'exemple, l'adresse d'interface Web est 132.209.24.59:8000, et 132.209.24.59 est l'adresse que le serveur crée, donc l'adresse varie selon l'ordinateur. Le nombre 8000 est le numéro de la porte définie dans la programmation.

La page porte le titre « Smart Home Control » et c'est la première zone que propose de capteur XBee. Cette partie indique d'abord l'adresse du capteur, ensuite la température, la lumière et l'humidité qu'il a détecté. Les informations des deux prises intelligentes sont

indiquées également à la différence que les prises intelligentes recueillent le courant d'électroménagers branchés plutôt que la donnée concernant l'humidité. D'ailleurs, le bouton radio est fourni pour contrôler les états des dispositifs XBee. Si un client clique sur « On » pour une prise donnée et soumet la commande, elle sera transmise à la prise afin de l'ouvrir pour que le courant puisse sortir. Le petit carré qui représente l'état de la prise va indiquer la couleur verte après avoir actualisé la page Web. En revanche, si la prise est fermée, le carré sera rouge.

The screenshot shows a web browser window with the address bar displaying '132.209.24.59:8000'. The page title is 'Smart Home Control'. The interface is divided into three main sections, each with a light blue header bar.

Section 1 (Sensor):

- Sensor addr is: 00:13:a2:00:40:ae:c2:5f
- Temperature (C): 24.123
- Light (Lux): 272.141
- Humidity (RH): 80.002

Section 2 (Smart plug_1):

- Smart plug_1 addr is: 00:13:a2:00:40:6e:41:61
- Temperature (C): 23.067
- Light (Lux): 761.29
- Current (A): -0.107
- Control(On/Off): ☒ On ☐ Off
- Submit

Section 3 (Smart plug_2):

- Smart plug_2 addr is: 00:13:a2:00:40:6e:3f:49
- Temperature (C): 21.308
- Light (Lux): 77.419
- Current (A): -0.246
- Control(On/Off): ☒ On ☐ Off
- Submit

Figure 5.19 L'interface web d'application interopérable dans HEMS

5.5 Conclusion

Dans ce chapitre, nous avons tout à bord cité et décrit les produits et les logiciels qui sont utilisés dans nos expérimentations. Étant donné que les produits de la compagnie Digi visent l'utilisation de la technologie ZigBee et que ses produits sont accessibles pour tous les développeurs, nous avons employé tous les produits ZigBee fournis par Digi.

La première expérimentation propose un moyen de construire un réseau ZigBee afin de contrôler la température par les appareils dans le réseau. Le réseau consiste en une passerelle ZigBee, une prise intelligente branchée à un chauffage portable, et un ordinateur. Après avoir configuré la passerelle et la prise intelligente par une interface locale, le programme est exécuté par le logiciel offert par Digi, qui s'appelle ESP. Sur les résultats de l'expérimentation, nous avons trouvé que la communication avec ZigBee fonctionne bien dans un scénario de maison.

Ensuite, nous avons essayé de construire un réseau de SEP avec une passerelle SEP et les appareils de ZigBee. Bien que SEP soit compatible avec ZigBee, il faut aussi transformer les modules en SEP. La transformation s'effectue simplement en changeant le « firmware » des modules et en installant le certificat de SEP. Cependant, les appareils qui avaient été transformés en SEP n'ont pas réussi à se connecter avec la passerelle SEP.

La dernière expérimentation visait à développer une application Web pour l'interopérabilité du réseau ZigBee. Deux grandes parties de programmes ont été exécutées respectivement dans la passerelle et l'ordinateur hôte. Grâce à la technique « socket() », la communication entre les différents éléments s'est réalisée. Nous avons aussi utilisé HTML pour accomplir la page Web. De plus, WSGI a servi à combiner le serveur Web et HTML.

Sur la page Web, les informations du capteur et des deux prises ont été exposées, ainsi que la commande qui peut être envoyée à la prise intelligente destinée.

Chapitre 6 - Conclusion

Ce travail consistait à construire un réseau de ZigBee, analyser les performances de certains appareils ZigBee, développer une application Web afin de surveiller et contrôler un réseau résidentiel.

HEMS est une solution pour assurer l'économique d'énergie et la gestion énergétique grâce à un réseau domestique. Un HEMS intègre les technologies d'information, de communication et d'affichage, et il offre une interface aux consommateurs pour gérer efficacement leur consommation d'électricité. Notre travail s'est surtout concentré sur la connectivité dans le réseau de HEMS. Plusieurs appareils intelligents commerciaux existent et sont équipés de diverses technologies de communication. Le Wi-Fi est la technologie de communication sans fil la plus utilisée. Cependant, celle-ci consomme beaucoup d'électricité, de sorte qu'elle n'est pas appropriée pour les appareils qui sont alimentés par pile. Cette étude a également évalué les technologies de HomePlug, Bluetooth, Z-Wave et ZigBee. Ainsi, nous avons constaté que HomePlug profite des fils de puissance; par conséquent, il n'est pas nécessaire d'y installer de nouveaux fils, car elle est capable de transmettre les données à longue distance. Bluetooth est un choix approprié pour un réseau local, et pourtant un Bluetooth à réseau maximal possède huit appareils. De plus, Z-Wave et ZigBee sont similaires, mais ZigBee offre plus de produits commerciaux.

Plusieurs technologies de communication coexistent probablement dans un HEMS et, afin de connecter les appareils qui utilisent différentes technologies de communication, une

passerelle est essentielle pour HEMS. La passerelle s'occupe tout d'abord d'échanger les informations entre le réseau public et le réseau local. Ensuite, une passerelle installe le protocole d'application SEP2.0 qui est compatible avec plusieurs protocoles dans la couche physique.

En nous basant sur les connaissances théoriques, nous avons fait trois expérimentations pour analyser les performances de ZigBee, tester la faisabilité de SEP, ainsi que développer une interface Web visant à accomplir l'interopérabilité de HEMS avec ZigBee. Dans la première expérimentation, un simple réseau ZigBee a été établi. Par la programmation sur la passerelle ZigBee, une prise intelligente a été branchée sur un chauffage portable et contrôlée. Lorsque la température détectée était plus élevée que 25 degrés, une commande était envoyée à la prise intelligente et la prise arrêta immédiatement la source du chauffage. Conséquemment, la température a été maintenue à 25 degrés centigrades. La passerelle recevait les données de la prise toutes les 10 secondes et le nombre d'erreurs était négligeable.

Pour la deuxième expérimentation, nous avons essayé de construire un réseau de SEP avec une passerelle SEP et les appareils de ZigBee. Tous les appareils dans un réseau SEP doivent être installés selon un certificat spécifique. Après la mise à jour du « firmware » et l'installation du certificat sur les appareils ZigBee, la connectivité n'a pas toujours réussi.

Pour la dernière expérimentation, nous avons développé une application Web qui fournit une interface d'interopérabilité pour un HEMS. Le HEMS consiste en un capteur ZigBee, deux prises ZigBee connectées avec deux charges, une passerelle ZigBee, et un ordinateur hôte. Par exécuter un programme sur la passerelle, les données du capteur et des prises ont été recueillies dans la passerelle. La page Web a été développée par HTML et le

serveur Web a simplement été établi par WSGI. Enfin, la page Web développée présentait les états des appareils ZigBee en temps réel et le client pouvait contrôler les prises à distance, sur la page Web.

Ce travail a permis de faire la communication suivante dans une conférence internationale : « IEEE/PES Asia Pacific Power and Energy Engineering Conférence (APPEEC2016), Xi'an, Chine, Oct 25-28, 2016 ». L'article participé à cette conférence intitulé « Connectivity for Home Energy Management Applications », il a obtenu le prix du meilleur article étudiant à la conférence.

Bibliographie

- [1] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "A Survey on smart grid potential applications and communication requirements," *IEEE Trans. Ind. Informatics*, vol. 9, no. 1, pp. 28–42, 2013.
- [2] Normazlina Binti Mat Isa, Tan CheeWei, and Abd Halim Mohd Yatim, "Smart Grid Technology: Communications, Power Electronics and Control System," in *International Conference on Sustainable Energy Engineering and Application(ICSEEA)*, 2015, pp. 10–14.
- [3] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings, "Agent-Based Control for Decentralised Demand Side Management in the Smart Grid," in *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems - Innovative Applications Track (AAMAS 2011)*, Taipei, Taiwan, pp. 5–12.
- [4] S. Meiling, T. Steinbach, M. Duge, and T. C. Schmidt, "Consumer-oriented integration of smart homes and smart grids: A case for multicast-enabled Home Gateways?," *Proc. 2013 IEEE 3rd Int. Conf. Consum. Electron. - Berlin, ICCE-Berlin*, pp. 279–283, 2013.
- [5] Statistics Canada, "Households and the Environment: Energy Use," 2011.
- [6] B. Asare-Bediako, W. L. Kling, and P. F. Ribeiro, "Home energy management systems: Evolution, trends and frameworks," in *International Universities Power Engineering Conference (UPEC), 47th London*, 2012, pp. 1–5.
- [7] L. Dan and H. Bo, "Advanced Metering Standard Infraxtructure for Smart Grid," *China Int. Conf. Electr. Distrib.*, no. Ciced, pp. 1–4, 2012.
- [8] "Overview and Comparison of Leading Communication Standard Technologies for Smart Home Area Networks Enabling Energy Management Systems," *46th Int. Univ. Power Eng. Conf.*, no. September, pp. 1–6, 2011.
- [9] N. Kushiro, S. Suzuki, M. Nakata, H. Takahara, and M. Inoue, "Integrated Residential Gateway Controller for Home Energy Management System," *IEEE Trans. Consum. Electron.*, vol. 49, pp. 629–636, 2003.
- [10] S. Darby, "the Effectiveness of Feedback on Energy Consumption a Review for Defra of the Literature on Metering , Billing and," no. April, 2006.

- [11] F. Bouhafs, M. Mackay, and M. Merabti, "Home Energy Management Systems," pp. 53–67, 2014.
- [12] "SEAI - Overview of the benefits of an energy management system."
- [13] T. Saito, I. Tomoda, Y. Takabatake, K. Teramoto, and K. Fujimoto, "Gateway technologies for home network and their implementations," in *Proceedings 21st International Conference on Distributed Computing Systems Workshops, Mesa, AZ*, 2001, no. 2, pp. 175–180.
- [14] R. Bolla, M. Giribaldi, R. Khan, and M. Repetto, "Design of home energy gateway boosting the development of smart grid applications at home," *2013 4th Annu. Int. Conf. Energy Aware Comput. Syst. Appl. ICEAC 2013*, pp. 103–108, 2013.
- [15] A. Rossello--Busquet, J. Soler, and L. Dittmann, "A Novel Home Energy Management System Architecture," *2011 UkSim 13th Int. Conf. Comput. Model. Simul.*, pp. 387–392.
- [16] N. Saito, "Ecological home network: An overview," *Proc. IEEE*, vol. 101, no. 11, pp. 2428–2435, 2013.
- [17] HomePlug Alliance, "About the HomePlug Alliance." [Online]. Available: <http://www.homeplug.org/alliance/alliance-overview>.
- [18] M. Z. Huq and S. Islam, "Home Area Network technology assessment for demand response in smart grid environment," in *Universities Power Engineering Conference (AUPEC), 2010 20th Australasian*, 2010, pp. 1–6.
- [19] Homeplug Powerline Alliance, "HomePlug AV White Paper," pp. 1–11, 2005.
- [20] HomePlug Alliance, "HomePlug AV2: Next-Generation Connectivity." [Online]. Available: http://www.homeplug.org/tech-resources/hpav2_next_gen/.
- [21] HomePlug Alliance, "HomePlug Green PHY: perfect fit for Smart Energy/Internet of Thing (IoT)." [Online]. Available: <http://www.homeplug.org/tech-resources/green-phy-iot/>.
- [22] HomePlug Alliance, "Why Use HomePlug." [Online]. Available: <http://www.homeplug.org/explore-homeplug/overview/>.
- [23] L. TP-Link Technologies Co., "Adapter with AC Pass Through." [Online]. Available: http://www.tp-link.com/en/products/details/cat-18_TL-PA4020P.html.
- [24] ZyXEL Communications Corp., "3-in-1 Wireless HomePlug AV Router," 2007. [Online]. Available: http://homeplug.certapp.net/media/files/certification/briefs/NBG-318S_1.pdf.
- [25] J. Jansons and T. Dorins, "Analyzing IEEE 802.11n standard: Outdoor

- performanace,” in *2nd International Conference on Digital Information Processing and Communications (ICDIPC)*, Klaipeda city, 2012, pp. 26–30.
- [26] B. Sidhu, H. Singh, and A. Chhabra, “Emerging Wireless Standards - WiFi, ZigBee and WiMAX,” *World Acad. Sci. Eng. Technol. Int. J. Electr. Comput. Energ. Electron. Commun. Eng.*, vol. 1, no. 11, pp. 42–48, 2007.
 - [27] Wi-Fi Alliance, “Wi-Fi CERTIFIED n,” 2016. [Online]. Available: <http://www.wi-fi.org/discover-wi-fi/wi-fi-certified-n>.
 - [28] D-Link, “mydlink Wi-Fi MOTION SENSOR (DCH-S150) Datasheet,” 2014.
 - [29] Honeywell International Inc., “Lyric Round Wi-Fi Thermostat – Second Generation (RCH9310WF),” 2015. [Online]. Available: <http://yourhome.honeywell.com/en/products/thermostat/lyric-thermostat>.
 - [30] S.-Y. Chiu, H.-P. Chang, and R.-C. Chang, “Providing mobile LAN access capability for Bluetooth devices,” *Ninth Int. Conf. Parallel Distrib. Syst. Proc.*, 2002.
 - [31] V. Abinayaa and A. Jayan, “Case Study on Comparison of Wireless Technologies in Industrial Applications,” *Int. J. Sci. Res. Publ.*, vol. 4, no. 2, pp. 2–5, 2014.
 - [32] J. Lee, Y. Su, and C. Shen, “A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” in *Industrial Electronics Society, IECON 2007. 33rd Annual Conference of the IEEE*, 2007, pp. 46–51.
 - [33] H. Fornazier, A. Martin, and S. Messner, “Wireless Communication: Wi-Fi, Bluetooth, IEEE 802.15.4, DASH7,” no. march, pp. 1–26, 2012.
 - [34] T. Zhang, J. Lu, F. Hu, and Q. Hao, “Bluetooth low energy for wearable sensor-based healthcare systems,” *2014 IEEE Healthc. Innov. Point-of-Care Technol. Conf. Seattle, WA*, pp. 251–254, 2014.
 - [35] Smartbotics, “Robosmart Wireless LED Light Bulb,” 2013. [Online]. Available: <http://www.smartbotics.com/#!/products/c218d>.
 - [36] ELgato, “Eve energy switch&meter.” [Online]. Available: <https://www.elgato.com/en/eve/eve-energy>.
 - [37] J. Cheng and T. Kunz, “A Survey on Smart Home Networking,” *Carlet. Univ. Syst. Comput. Eng. Tech. Rep. SCE-09-10*, 2009.
 - [38] M. T. Galeev, “Catching the z-wave,” *Electron. Eng. Times India*, no. Oct, 2006.
 - [39] G. Ferrari, P. Medagliani, S. Di Piazza, and M. Martalò, “Wireless sensor networks: Performance analysis in indoor scenarios,” *EURASIP J. Wirel. Commun. Netw.*, 2007.

- [40] Honeywell International Inc., "VISION PRO Z-WAVE THERMOSTAT," 2016. [Online]. Available: <http://www.z-wave.com/products/vision-pro-z-wave-thermostat>.
- [41] Leviton, "VRCS2-MRZ," 2016. [Online]. Available: http://french.leviton.com/OA_HTML/ProductDetail.jsp?partnumber=VRCS2-MRZ§ion=44147.
- [42] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Commun. Mag.*, vol. 48, no. 6, pp. 92–101, 2010.
- [43] D. Han and J. Lim, "Smart home energy management system using IEEE 802.15.4 and zigbee," *IEEE Trans. Consum. Electron.*, vol. 56, pp. 1403–1410, 2010.
- [44] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Review of communication technologies for smart homes/building applications," *2015 IEEE Innov. Smart Grid Technol. - Asia (ISGT ASIA)*, pp. 1–6, 2015.
- [45] D. Bian, M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Assessment of communication technologies for a home energy management system," *Innov. Smart Grid Technol. Conf. (ISGT), 2014 IEEE PES*, no. 19–22 Feb, 2014.
- [46] A. Hafeez, N. H. Kandil, B. Al-Omar, T. Landolsi, and A. R. Al-Ali, "Smart home area networks protocols within the smart grid context," *J. Commun.*, vol. 9, no. 9, pp. 665–671, 2014.
- [47] "Mct--340 sma Door / Window Sensor, Installation Instructions," *Visonic*, 2014. [Online]. Available: https://support.smarthings.com/hc/en-gb/article_attachments/201896836/Tyco_MCT-340_Manual.pdf.
- [48] "Wireless Area Controller – WAC50," *Daintree Networks, Inc.*, 2012. [Online]. Available: http://www.daintree.net/wp-content/uploads/2014/04/wac50_cutsheet.pdf.
- [49] M. Zareei, A. Zarei, R. Budiarto, and M. A. Omar, "A Comparative Study of Short Range Wireless Sensor Network on High Density Networks," *17th Asia-Pacific Conf. Commun.*, no. October, pp. 247–252, 2011.
- [50] Texas Instruments, "Smart Energy Profile Solutions," 2011.
- [51] IEEE Standards Association, "IEEE Adoption of Smart Energy Profile 2 . 0 Application Protocol Standard," 2013.
- [52] Robby Simpson, "An Overview of Smart Energy Profile 2 . 0," *GE Digital Energy*, 2013. [Online]. Available: hes-standards.org/doc/SC25_WG1_N1612.pdf.
- [53] Lee H. Goldberg, "ZigBee's Smart Energy 2.0 Profile Brings New Capabilities and Design Challenges," *Digi-Key, ArticleLibrary*, 2014. [Online]. Available: <http://www.digikey.com/en/articles/techzone/2014/oct/zigbees-smart-energy-20->

profile-brings-new-capabilities-and-design-challenges.

- [54] Manas Saksena, "Design How To Smart Energy Profile (SEP) 2.0 Uncovered," 2011. [Online]. Available: http://www.eetimes.com/document.asp?doc_id=1279156.
- [55] Certicom corps. , "No CERTICOM DEVICE CERTIFICATION AUTHORITY FOR ZIGBEE SMART ENERGY." [Online]. Available: <https://www.certicom.com/products-and-services/managed-certificate-service/certicom-security-for-zigbee-smart-energy>.
- [56] Zigbee Alliance, "ZigBee Smart Energy." [Online]. Available: <http://www.zigbee.org/zigbee-for-developers/applicationstandards/zigbeesmartenergy/>.
- [57] NETGEAR, "Home lifestyle gateway." [Online]. Available: https://www.netgear.com/service-providers/products/home-monitoring-automation/gateways/asgl100.aspx?cid=wmt_netgear_organic#tab-features.
- [58] Digi International, "About Digi." [Online]. Available: <http://www.digi.com/aboutdigi>.
- [59] Digi International, "Digi Xbee Smart Plug," 2015. [Online]. Available: http://www.digi.com/pdf/ds_digixbeesmartplug.pdf.
- [60] Digi International, "XBee Drop-in-Networking Accessories User's Guide," 2015. [Online]. Available: <ftp1.digi.com/support/documentation/90000891.pdf>.
- [61] Digi International, "XBee ® Gateway Family." [Online]. Available: www.digi.com/pdf/ds_xbeegateway.pdf.
- [62] Digi International, "Feature Spec Digi ESP ™," 2014. [Online]. Available: www.digi.com/pdf/fs_esp.pdf.
- [63] Digi International, "Digital and analog sampling using XBee radios," 2015. [Online]. Available: http://knowledge.digi.com/articles/Knowledge_Base_Article/Digital-and-analog-sampling-using-XBee-radios.
- [64] Digi International, "ConnectPort X2e® for Smart Energy." [Online]. Available: <http://www.digi.com/products/xbee-rf-solutions/gateways/connectportx2ese>.
- [65] Certicom corps. , "certicom." [Online]. Available: <https://www.certicom.com/index.php/gencertregister>.
- [66] Python Software Foundation, "Python2.7.12 Documentation, The Python Standard Library, 16.1.select-Waiting for I/O completion," 2016. [Online]. Available: <https://docs.python.org/2/library/select.html>.
- [67] Y. Luo, "Using JavaScript to Implement the Real Time Feedback in Online

- Learning,” *J. Eng. Technol. Educ.*, vol. 7, no. 1, pp. 16–20, 2010.
- [68] J. Korpela, “Lurching Toward Babel: HTML, CSS, and XML,” *Computer (Long Beach, Calif.)*, pp. 103–106, 1998.
- [69] A. Holovaty, S. Willison, J. Kaplan-Moss, and W. Miner, “Django Documentation,” pp. 1–1172, 2016.
- [70] Python Software Foundation, “Python2.7.12 Documentation, The Python Standard Library, 20.4 wsgiref-WSGI Utilities and Reference Implementation,” 2016. [Online]. Available: <https://docs.python.org/2/library/wsgiref.html?highlight=wsgi#module-wsgiref>.

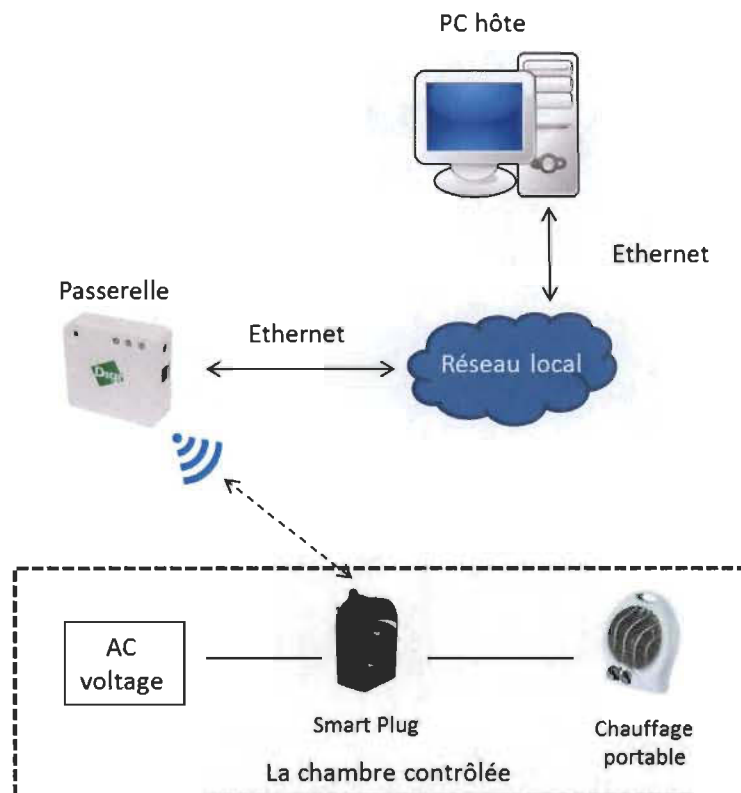
Annexe A – Manuel d'utilisation

a. Expérimentation pour contrôler le « SmartPlug » par la température

1. Travail préparatoire

1) L'installation du matériel

La passerelle se connecte au PC directement ou par Ethernet, puis se connecte aux produits de Xbee par la technologie de Zigbee.



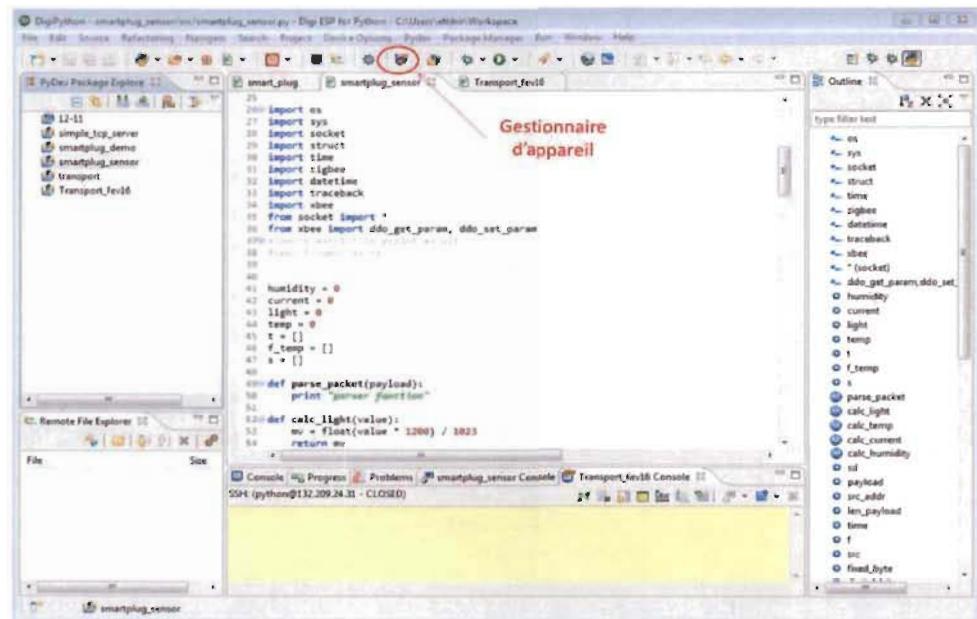
2) Télécharger et installer Digi ESP

L'adresse pour télécharger Digi ESP se trouve ci-dessous :

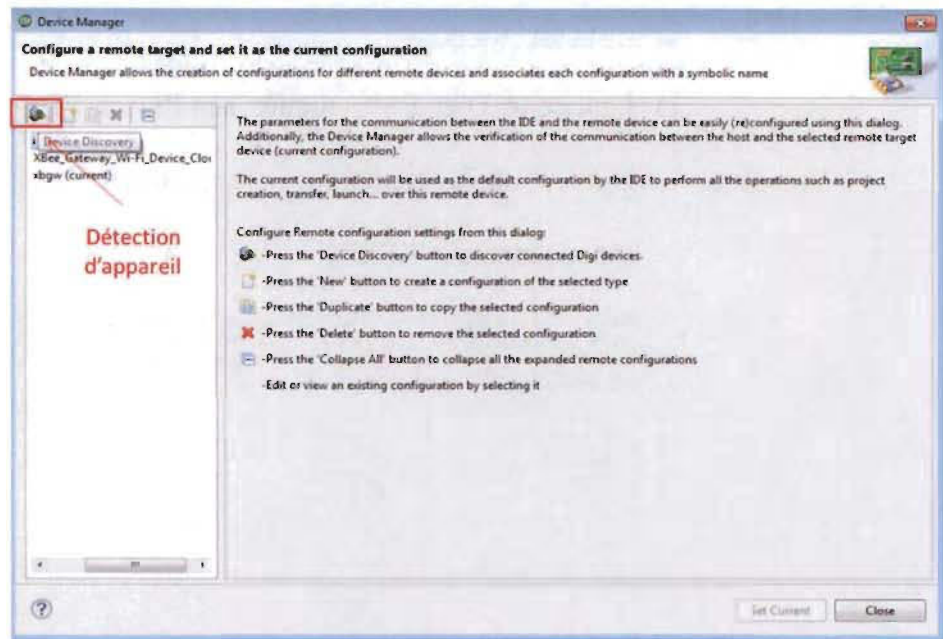
<http://www.digi.com/products/xbee-rf-solutions/gateways/xbee-gateway#productsupport>

2. Construction du réseau

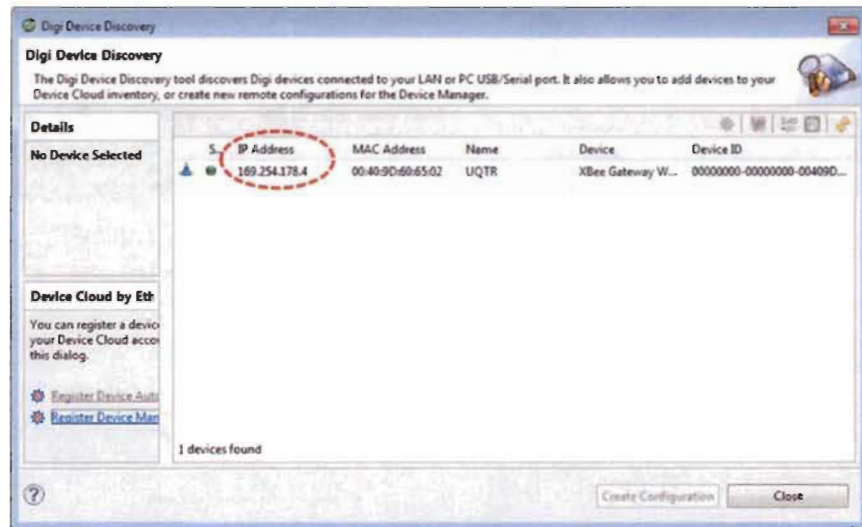
- 1) Lancez le Digi ESP, cliquez sur l'icône de gestionnaire d'appareil dans la barre d'outils (remarque : le pare-feu de l'ordinateur doit autoriser le fonctionnement d'ESP.)



- 2) Sur l'interface de dialogue du gestionnaire d'appareil, cliquez sur l'icône de la détection d'appareil. La passerelle peut être détectée soit par le réseau local soit par l'infonuagique de Digi, car la connexion se fait par le réseau. Dans nos expérimentations, nous avons choisi le réseau local pour trouver la passerelle XBee connectée.



Si une interface affiche les informations de la passerelle, c'est parce que la passerelle est bien connectée avec l'ordinateur, ils sont dans le même réseau local. Sur cette interface, nous pouvons trouver l'adresse IP de la passerelle.

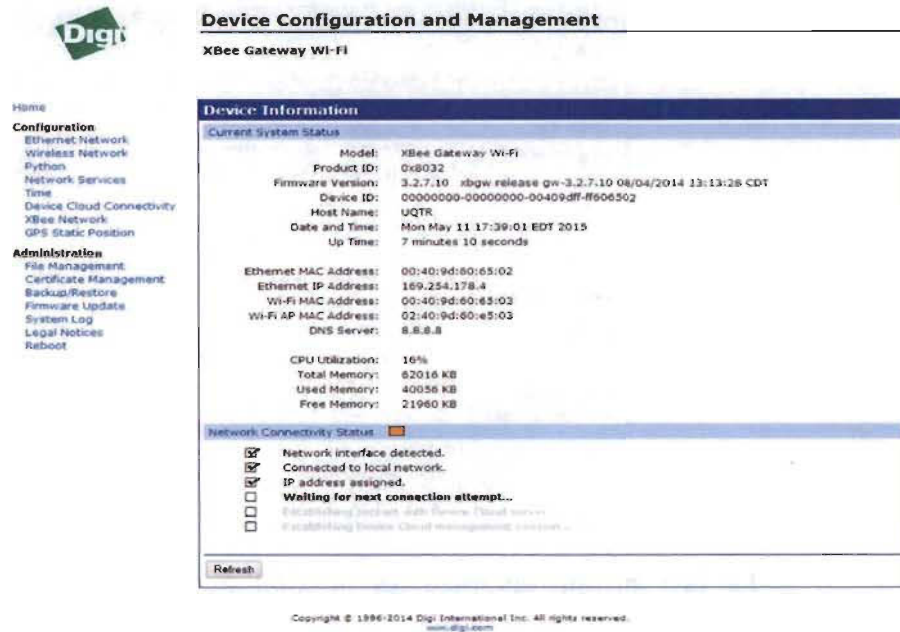


Le lien suivant énumère les possibilités de problèmes et les solutions pour la détection des appareils.

http://knowledge.digi.com/articles/Knowledge_Base_Article/Device-Discovery-Troubleshooting-Tips

3. Configuration

- 1) L'adresse IP de la passerelle est aussi l'adresse de l'interface Web locale. Cette interface Web, développée par Digi, est conçue pour observer les connexions et configurer les dispositifs des matériels. Ouvrez l'interface Web locale par le navigateur.



Device Configuration and Management
XBee Gateway Wi-Fi

Home
Configuration
Ethernet Network
Wireless Network
Python
Network Services
Time
Device Cloud Connectivity
XBee Network
GPS Static Position
Administration
File Management
Certificate Management
Backup/Restore
Firmware Update
System Log
Legal Notices
Reboot

Device Information
Current System Status

Model: XBee Gateway Wi-Fi
Product ID: 0x8032
Firmware Version: 3.2.7.10 xbgw release gw-3.2.7.10 08/04/2014 13:13:26 CDT
Device ID: 00000000-00000000-00409dff-ff606502
Host Name: UQTR
Date and Time: Mon May 11 17:39:01 EDT 2015
Up Time: 7 minutes 10 seconds

Ethernet MAC Address: 00:40:9d:60:65:02
Ethernet IP Address: 169.254.178.4
Wi-Fi MAC Address: 00:40:9d:60:65:03
Wi-Fi AP MAC Address: 02:40:9d:60:e5:03
DNS Server: 8.8.8.8

CPU Utilization: 16%
Total Memory: 62016 KB
Used Memory: 40056 KB
Free Memory: 21960 KB

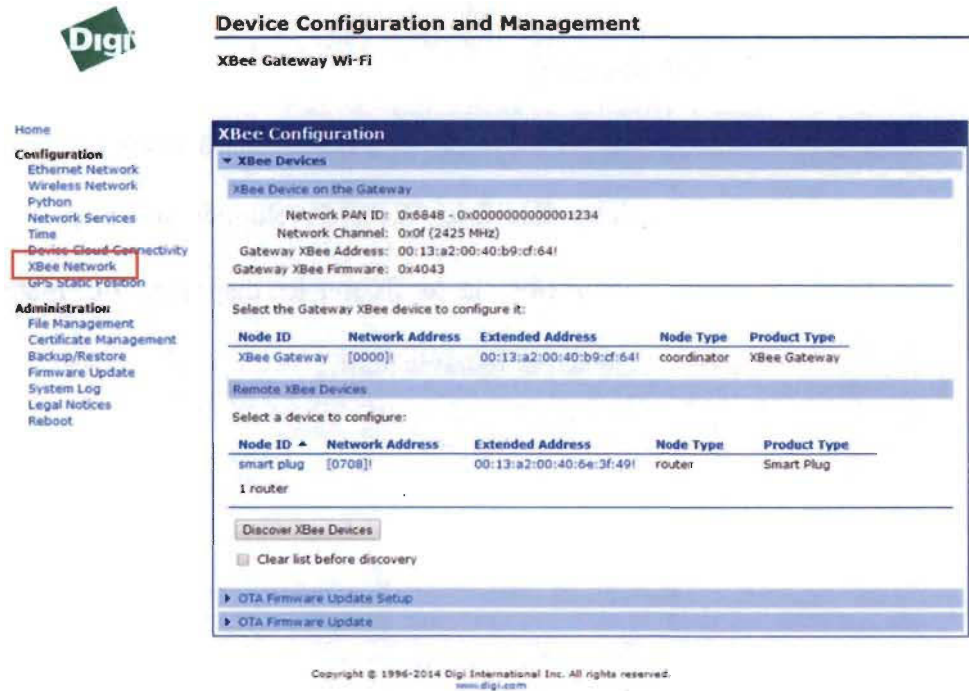
Network Connectivity Status

☒ Network interface detected.
☒ Connected to local network.
☒ IP address assigned.
☐ Waiting for next connection attempt...
☐ Establishing contact with remote cloud server
☐ Establishing remote cloud management session...

Refresh

Copyright © 1996-2014 Digi International Inc. All rights reserved.
www.digi.com

- 2) Sous le menu de configuration, la partie « XBee Network » offre une interface du réseau XBee. Sur la page, la passerelle connectée et les appareils trouvés par la passerelle sont présentés.



Device Configuration and Management
XBee Gateway Wi-Fi

Home
Configuration
Ethernet Network
Wireless Network
Python
Network Services
Time
Device Cloud Connectivity
XBee Network
GPS Static Position
Administration
File Management
Certificate Management
Backup/Restore
Firmware Update
System Log
Legal Notices
Reboot

XBee Configuration

XBee Devices

XBee Device on the Gateway

Network PAN ID: 0x6848 - 0x00000000000001234
Network Channel: 0x0f (2425 MHz)
Gateway XBee Address: 00:13:a2:00:40:b9:cf:64f
Gateway XBee Firmware: 0x4043

Select the Gateway XBee device to configure it:

| Node ID | Network Address | Extended Address | Node Type | Product Type |
|--------------|-----------------|--------------------------|-------------|--------------|
| XBee Gateway | [0000] | 00:13:a2:00:40:b9:cf:64f | coordinator | XBee Gateway |

Remote XBee Devices

Select a device to configure:

| Node ID | Network Address | Extended Address | Node Type | Product Type |
|------------|-----------------|--------------------------|-----------|--------------|
| smart plug | [0708] | 00:13:a2:00:40:6e:3f:49f | router | Smart Plug |

1 router

Discover XBee Devices
☐ Clear list before discovery

OTA Firmware Update Setup
OTA Firmware Update

Copyright © 1996-2014 Digi International Inc. All rights reserved.
www.digi.com

- 3) Double cliquer sur l'appareil qui a besoin de configuration. La page de configuration est alors comme la figure suivante. Modifier les données d'AT

D1, D2, D3 à 2 pour permettre au capteur de SmartPlug de fonctionner. IR représente le taux des données, par exemple : IR=10s; la passerelle va recevoir l'ensemble des données toutes les 10 secondes. Il faut faire attention, car la valeur de PR (pull-up) peut affecter la donnée de courant détectée par SmartPlug.



Device Configuration and Management

XBee Gateway Wi-Fi

Home

Configuration

Ethernet Network
Wireless Network
Python
Network Services
Time
Device Cloud Connectivity
XBee Network
GPS Static Position

Administration

File Management
Certificate Management
Backup/Restore
Firmware Update
System Log
Legal Notices
Reboot

XBee Configuration

[Return to Network View](#)

Device Details

Node ID: smartplug2
Extended Address: 00:13:a2:00:40:6e:3f:d9
Product Type: Smart Plug
Firmware Version: 0x29a7

Network Settings

Addressing Settings

Radio Settings

Serial Settings

Input/Output Settings

I/O Pin Settings

| DIO | AT | Functions | Setting | Pull up | Detect |
|-----|----|---------------|--------------------------|--------------------------|-------------------------------------|
| 0 | D0 | AD0, CB | Commissioning Button (1) | <input type="checkbox"/> | <input type="checkbox"/> |
| 1 | D1 | AD1 | Analog Input (2) | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 2 | D2 | AD2 | Analog Input (2) | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3 | D3 | AD3 | Analog Input (2) | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 4 | D4 | | Digital Output, Low (4) | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 5 | D5 | Associate LED | Associated Indicator (1) | <input type="checkbox"/> | <input type="checkbox"/> |
| 6 | D6 | RTS | Disabled (0) | <input type="checkbox"/> | <input type="checkbox"/> |
| 7 | D7 | CTS, RS-485 | CTS Flow Control (1) | <input type="checkbox"/> | <input type="checkbox"/> |
| 10 | P0 | RSSI PWM | RSSI PWM Output (1) | <input type="checkbox"/> | <input type="checkbox"/> |
| 11 | P1 | | Disabled (0) | <input type="checkbox"/> | <input type="checkbox"/> |
| 12 | P2 | | Disabled (0) | <input type="checkbox"/> | <input type="checkbox"/> |

Pull-up Resistor Enable (PR): 0x80 bitfield (0-0x7fff)

DIO Change Detect (IC): 0x1e bitfield (0-0xffff)

I/O Settings

Associate LED Blink Time (LT): 0 x 10 msec (10-255, 0=default)

RSSI PWM Timer (RP): 40 x 100 msec (0-255)

I/O Sampling Rate (IR): 10000 msec (32-65535, 0=disabled)

Supply Voltage High Threshold (V+): 0 mvolts (0-65535)

[Apply](#) [Refresh](#)

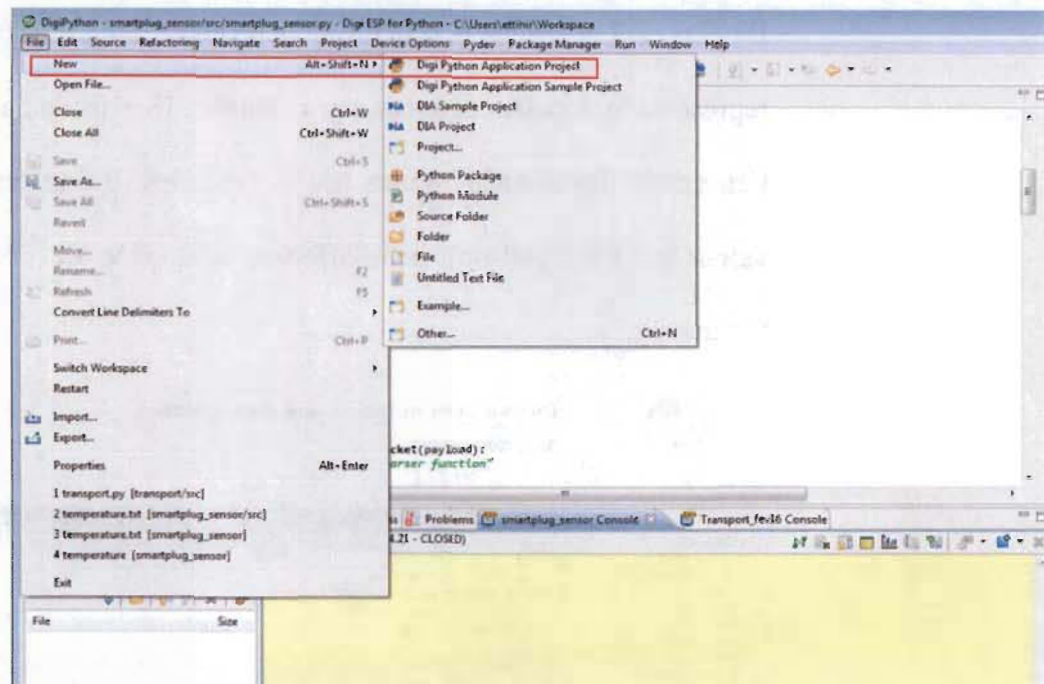
Device Status

Device Operations

4. Développement de projet

Démarrer un nouveau projet vide sur ESP. Écrire notre programme dans le projet.

ESP >> File >> New >> Digi Python application projet.

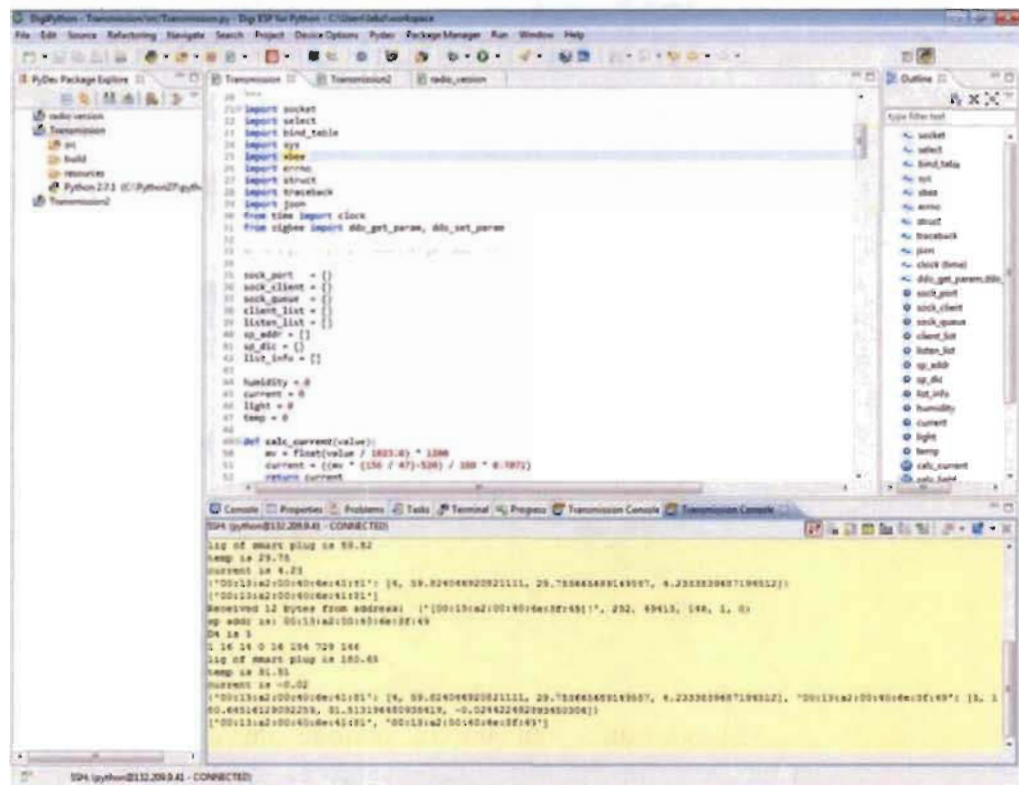
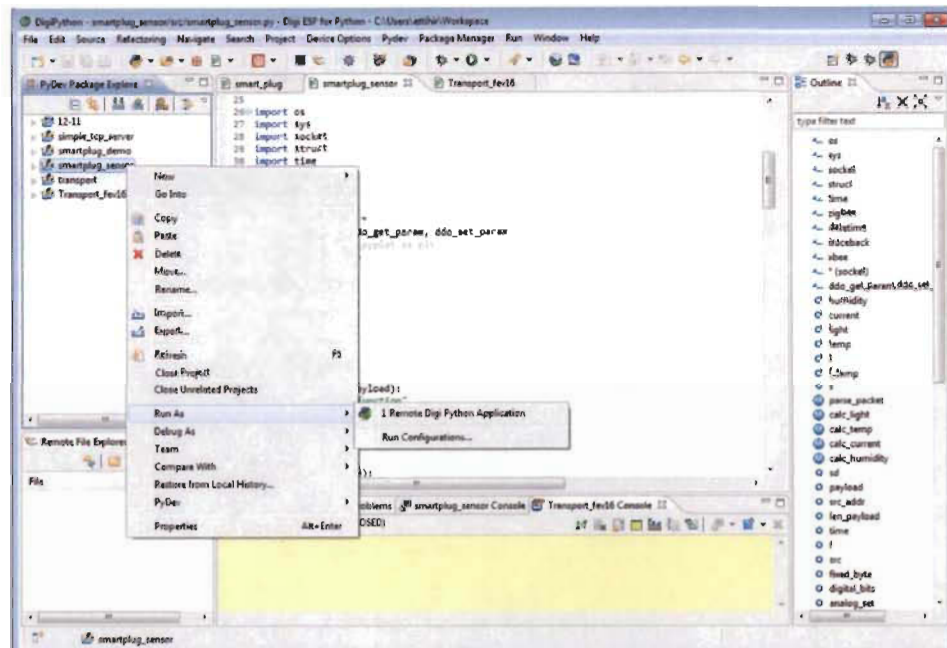


5. Exécution de programmation

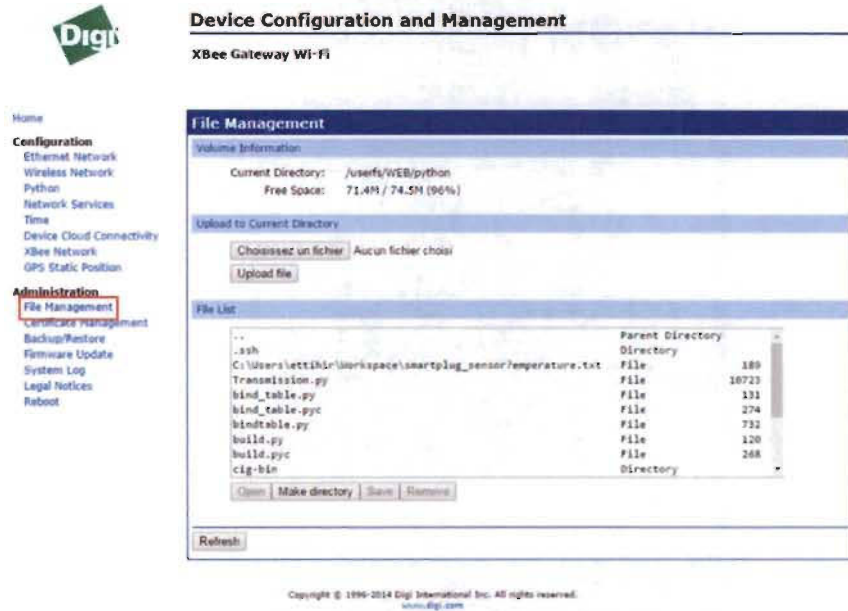
Deux moyens sont possibles pour exécuter le programme sur la passerelle :

1) Exécuter le programme sur ESP

En cliquant avec le bouton droit de la souris sur le nom de projet que nous voulons exécuter, le programme à la passerelle connecté s'exécute. S'il existe des erreurs dans le programme, les messages d'erreurs et les débogages seront présentés dans l'espace de la console (la partie jaune). Par contre, dans la console, les résultats du programme seront présentés.



- 2) Sauvegarder le programme vers la passerelle par l'interface Web locale
 Enregistrez le programme sous un fichier python (.py). Dans l'interface Web locale de configuration, puis cliquez « file management » dans la liste du menu pour ouvrir une page afin de gérer les fichiers dans la passerelle. Ensuite, enregistrez le fichier de script Python sur la page.



Device Configuration and Management
XBee Gateway Wi-Fi

Home

Configuration

- Ethernet Network
- Wireless Network
- Python
- Network Services
- Time
- Device Cloud Connectivity
- XBee Network
- GPS Static Position

Administration

- File Management**
- Certificate Management
- Backup/Restore
- Firmware Update
- System Log
- Legal Notices
- Reboot

File Management

Volume Information

Current Directory: /users/WEB/python
Free Space: 71.4M / 74.5M (96%)

Upload to Current Directory

Choisissez un fichier | Aucun fichier choisi

Upload file

File List

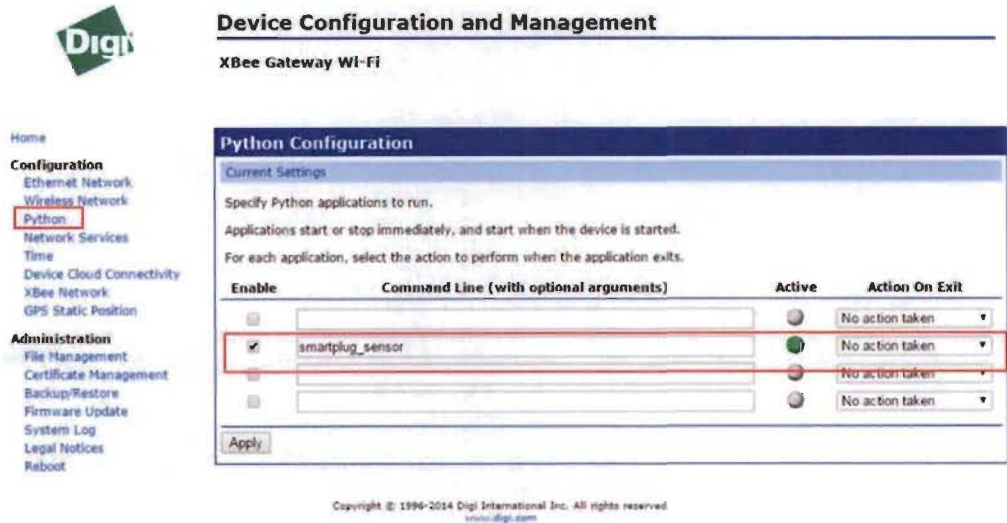
| File | Parent Directory | Size |
|---|------------------|-------|
| .. | Directory | |
| ..ssh | Directory | |
| C:\Users\lettikr\workspace\smartplug_sensor\temperature.txt | File | 189 |
| Transmission.py | File | 10723 |
| bind_table.py | File | 131 |
| bind_table.pyc | File | 274 |
| bindtable.py | File | 732 |
| build.py | File | 120 |
| build.pyc | File | 268 |
| cig-bin | Directory | |

Open Make directory Save Remove

Refresh

Copyright © 1999-2014 Digi International Inc. All rights reserved.
www.digi.com

Ouvrez la page de la configuration Python. Inscrire le nom de fichier à enregistrer et le cocher.



Device Configuration and Management
XBee Gateway Wi-Fi

Home

Configuration

- Ethernet Network
- Wireless Network
- Python**
- Network Services
- Time
- Device Cloud Connectivity
- XBee Network
- GPS Static Position

Administration

- File Management
- Certificate Management
- Backup/Restore
- Firmware Update
- System Log
- Legal Notices
- Reboot

Python Configuration

Current Settings

Specify Python applications to run.
Applications start or stop immediately, and start when the device is started.
For each application, select the action to perform when the application exits.

| Enable | Command Line (with optional arguments) | Active | Action On Exit |
|-------------------------------------|--|----------------------------------|-----------------|
| <input type="checkbox"/> | | <input type="radio"/> | No action taken |
| <input checked="" type="checkbox"/> | smartplug_sensor | <input checked="" type="radio"/> | No action taken |
| <input type="checkbox"/> | | <input type="radio"/> | No action taken |
| <input type="checkbox"/> | | <input type="radio"/> | No action taken |

Apply

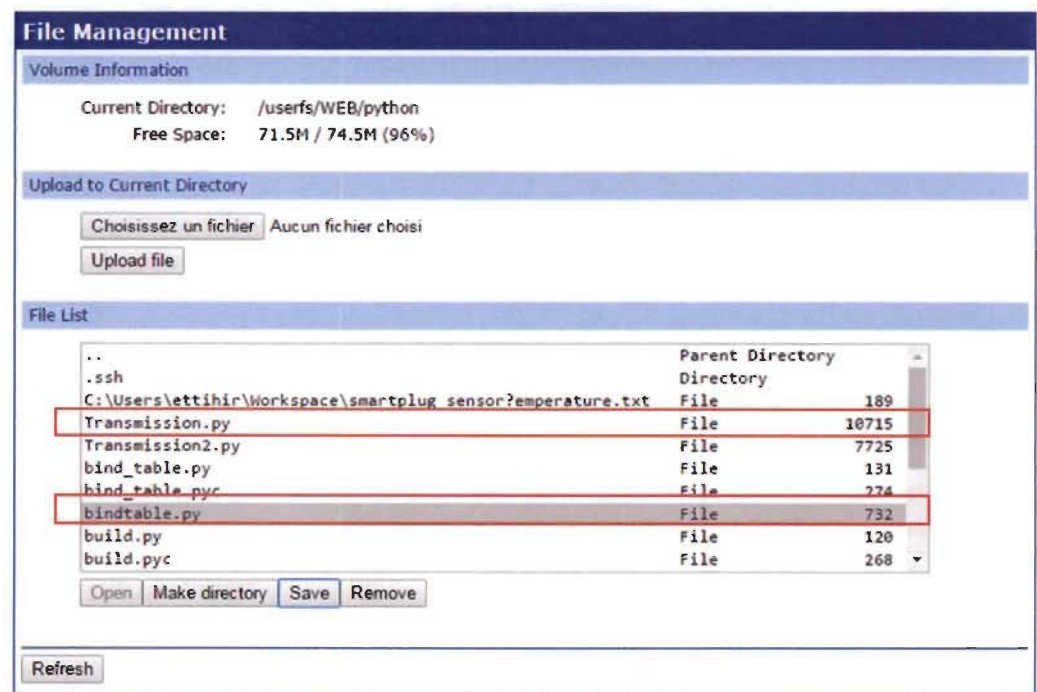
Copyright © 1999-2014 Digi International Inc. All rights reserved.
www.digi.com

Suggestion : Pendant la période de développement et de vérification de programme, il est préférable d'exécuter le programme par ESP. L'autre façon est plus appropriée quand le programme a été débogué et qu'il est nécessaire de toujours l'exécuter dans la passerelle.

b. Le développement de l'application Web pour contrôler et surveiller le réseau XBee

Parce que cette expérimentation comprend les programmes exécutés sur la passerelle et le programme exécuté dans l'ordinateur. Le processus qui exécute les programmes sur la passerelle est le même que pour l'expérimentation (a), sauf que dans l'étape 5.

- 1) Donc, il faut d'abord enregistrer le fichier « bindtable.py » sur l'interface Web locale.
- 2) Et plus, il faut enregistrer le programme principal qui est responsable de gérer les appareils XBee dans la passerelle.



- 3) L'autre programme qui comprend le code de HTML et le serveur est enregistré dans l'ordinateur sous le fichier Python (.py). Ouvrez ce programme par IDLE (IDLE est un GUI pour le programme Python, et il a été installé automatiquement avec ESP).
- 4) Exécutez le programme dans l'ordinateur (presser F5). Indiquez l'adresse IP de l'ordinateur avec la porte définie dans le navigateur (ex.: 132.209.9.55).

The screenshot shows a web browser window with the address bar displaying '132.209.9.55:8000'. The main content area is titled 'Smart Home Control' and contains three sections for controlling smart devices. Each section has input fields for sensor address, temperature, light, and humidity, and a 'Submit' button. The first section is for 'Smart plug_1' with address '00:13:a2:00:40:5e:41:81'. The second section is for 'Smart plug_2' with address '00:13:a2:00:40:5e:3f:49'. The third section is for 'Smart plug_3' with address '00:13:a2:00:40:5e:3e:49'. The 'Control/On Off' section shows 'On' selected. The terminal window on the right displays the following code:

```

File Edit Shell Debug Options Windows Help
[00/00/2016 16:10:20] "GET / HTTP/1.1" 200 5427
QUERY_STRING:
REQUEST_METHOD: GET
# connected
["00:13:a2:00:40:5e:41:81", "00:13:a2:00:40:5e:3e:49"]
smartplug1 addr 1a: 00:13:a2:00:40:5e:41:81
smartplug2 addr 1a: 00:13:a2:00:40:5e:3e:49
+ 48.031304158 29.982487947 4.2303034872
5 154.838709477 31.5131964809 0.00223451094819
[]
[00/00/2016 16:10:21] "GET / HTTP/1.1" 200 5427
QUERY_STRING:
REQUEST_METHOD: GET
# connected
["00:13:a2:00:40:5e:41:81", "00:13:a2:00:40:5e:3e:49"]
smartplug1 addr 1a: 00:13:a2:00:40:5e:41:81
smartplug2 addr 1a: 00:13:a2:00:40:5e:3e:49
+ 48.031304158 29.982487947 4.2303034872
5 154.838709477 31.5131964809 0.00223451094819
[]
[00/00/2016 16:10:22] "GET / HTTP/1.1" 200 5428
QUERY_STRING:
REQUEST_METHOD: GET
# connected
["00:13:a2:00:40:5e:41:81", "00:13:a2:00:40:5e:3e:49"]
smartplug1 addr 1a: 00:13:a2:00:40:5e:41:81
smartplug2 addr 1a: 00:13:a2:00:40:5e:3e:49
+ 48.031304158 29.982487947 4.2303034872
5 154.838709477 31.5131964809 0.00223451094819
[]
[00/00/2016 16:10:23] "GET / HTTP/1.1" 200 5428

```

Annexe B – Code Source des Expérimentations

- a) Code source d'expérimentations à propos de contrôler un SmartPlug selon la température :

```
#####
# Ce programme se réfère aux exemples sur le site web de Digi
# Voici le lien d'exemple:
# http://www.digi.com/wiki/developer/index.php/Smart\_Plug\_Interactive\_Demo
#####
import os
import sys
import struct
import time
import zigbee
import datetime
import traceback
import xbee
from socket import *
from xbee import ddo_get_param, ddo_set_param

#####
# Initialiser les valeurs de l'humidité, le courant, la lumière et la température
#####
humidity = 0
current = 0
light = 0
temp = 0

t = []    # liste du temps quand reçu les données
```



```

f_temp = [] # Liste des températures détectées par SmartPlug
s = []      # Liste des états de SmartPlug

#####
# Définir les équations pour transformer les valeurs analogiques aux nombres
# compréhensibles
#####
def calc_light(value):
    mv = float(value * 1200) / 1023
    return mv

def calc_temp(value):
    mv = float(value * 1200) / 1023
    degc = ((mv - 500.0) / 10.0) - 4
    #degf = (degc * 1.8) + 32.0
    return degc

def calc_current(value):
    mv = float(value / 1023.0) * 1200
    current = ((mv * (156 / 47) - 520) / 180 * 0.7071)
    return current

def calc_humidity(value):
    mv = float((value / 1023) * 1200) * 108.2 / 33.2
    humidity = (mv / 5000 - 0.16) / 0.0062
    return humidity

#####
# Créer un socket afin d'établir la connexion avec SmartPlug
#####
try:
    sd = socket(AF_XBEE, SOCK_DGRAM, XBS_PROT_TRANSPORT)

```

```

#Bind à ("", 0xe8, 0xc105, 0x92) pour recevoir les réponses de 'IS' et les décoder
sd.bind((" ", 0xe8, 0xc105, 0x92))
print "socket is bound"

while 1:
#####
#Recevoir les données de SmartPlug circulaire, et enregistrer les certaines données
#que nous avons besoin en .txt. Les données vont entrer à MATLAB pour générer
#les figures plot.
#####
    try:
        payload, src_addr = sd.recvfrom(255) # Recevoir le package de données et
                                                # adresse de SmartPlug
        len_payload = len(payload)           # Longueur de données

        print datetime.datetime.now()

        #####
        # Écrire le temps à un fichier .txt
        #####
        time = str(datetime.datetime.now())
        t.append(time + '\r\n')               # Écrire le temps à la liste t
        print t

        f = open("Time.txt", "w")             # Ouvrir le fichier Time.txt
        f.writelines(t)                       # Écrire t à Time.txt
        f.close()                             # Fermer Time.txt

        #####
        # Imprimer l'adresse de SmartPlug
        #####
        src = src_addr[0][1:24]

```

```

print "Source Address: %s" %src

#####
# Déterminer l'appareil est SmartPlug
#####
if len_payload == 12:
    print len_payload
    fixed_byte, digital_bits, analog_set, dont_know_bit1, dont_know_bit2, lig,
    tmp, current = struct.unpack(">bhbhbhhh", payload[:12])
    light = calc_light(lig)
    temp = calc_temp(tmp)
    current = calc_current(current)

    print "temp is %4.2f" %temp
    #print "light is %4.2f" %light
    #print "current is %4.2f" %current
    #print "humidity is %4.2f" %humidity

    #####
    #Écrire la température à un fichier .txt
    #####
    f_temp.append(str(temp)+'\r\n')      # Écrire le température à la liste f_temp
    print f_temp
    f = open("Temperature.txt", "w")      # Ouvrir le fichier Température.txt
    f.writelines(f_temp)                  # Écrire f_temp à Température.txt
    f.close()                             # Fermer Température.txt

else:
    print "invalid payload"
    print len_payload

```

```

#####
# Contrôler le SmartPlug dépend de la température
#####
if temp > 25.00:
    ddo_set_param(src, 'D4', 4)      # 'D4' contrôle le courant de sortir de
    print "smartplug close"         # SmartPlug. Lorsque la température plus
                                    # que 25 degrés, configurer la valeur de 'D4'
                                    # à 4, SmartPlug va couper le courant.

    #####
    # Écrire l'état au fichier .txt
    #####
    s.append('0'+'\r\n')             # 0 représente le SmartPlug est fermé.
    print s                          # Écrire l'état de SmartPlug à la liste s
    f = open("State.txt", "w")       # Ouvrir le fichier State.txt
    f.writelines(s)                  # Écrire s à State.txt
    f.close()                        # Fermer State.txt
else:
    ddo_set_param(src, 'D4', 5)      # Lorsque la température moins que 25
degrés,
    print "smartplug open"           # configurer la valeur de 'D4' à 5, SmartPlug
                                    # va fournir l'électricité pour charge.

    #####
    # Écrire l'état au fichier .txt
    #####
    s.append('1'+'\r\n')             # 1 représente le SmartPlug est ouvert.
    print s                          # Écrire l'état de SmartPlug à la liste s
    f = open("State.txt", "w")       # Ouvrir le fichier State.txt
    f.writelines(s)                  # Écrire s à State.txt
    f.close()                        # Fermer State.txt

except Exception, e:

```

```

        print "Exception %s" %e
        traceback.print_exc()
except socket.error, s:
    print "socket exception: %s" %s

sd.close()                                # Fermer socket
print "socket is closed"

```

b) Code source d'expérimentations à propos du développement d'une application web

i. Détecter les nœuds connectés avec la passerelle.

```

import xbee

stored_nodes = {}                        # Un dictionnaire pour enregistrer les
                                         # adresses des nœuds

port = 4000
output_file = "bind_table.py"           # Le dictionnaire enregistre dans un fichier
                                         # « bind_table.py »

retries = 5

for i in xrange(0, retries):

    new_nodes = xbee.getnodelist()       # Obtenir les adresses et les portes des nœuds

    for node in new_nodes:
        if node.addr_extended not in stored_nodes: # Déterminer s'il y a de
                                                    # nouveaux nœuds
            stored_nodes[node.addr_extended] = port # S'il existe de nouveaux nœuds,
            port += 1                               # les ajouter dans le dictionnaire
                                                    # stored_nodes

    fh = open("/userfs/WEB/python/%s" %output_file, 'w') # Ouvrir le fichier
                                                         # « bind_table.py »
    fh.write("node_list = {\r\n")                  # Écrire les adresses et
                                                         # leur porte au fichier.

    key_list = stored_nodes.keys()
    for key in key_list:
        if key == key_list[-1]:

```

```

        fh.write("\t \"%s\":%d\r\n" %(key, stored_nodes[key]))
    else:
        fh.write("\t \"%s\":%d,\r\n" %(key, stored_nodes[key]))

    fh.write("}\r\n")
    fh.close()                                     #Fermer le fichier

print "Wrote %d entries" %len(key_list)

```

- ii. Le principal programme pour établir la connexion de XBee et TCP

```

import socket
import select
import bind_table
import sys
import xbee
import errno
import struct
import traceback
from time import clock
from zigbee import ddo_get_param, ddo_set_param

#bind_args, xb_psize = xbee_info.get_xbee_info()

#####
# Initialiser les valeurs de l'humidité, le courant, la lumière et la température
#####
sock_port = {}
sock_client = {}
sock_queue = {}
client_list = []
listen_list = []
xbee_addr = []
xbee_dic = {}
list_info = []

humidity = 0
current = 0
light = 0
temp = 0

```

```
#####
# Les équations pour transformer les valeurs analogiques aux nombres
# compréhensibles
#####
def calc_current(value):
    mv = float(value / 1023.0) * 1200
    current = ((mv * (156 / 47) - 520) / 180 * 0.7071)
    return current

def calc_light(value):
    mv = float(value * 1200) / 1023
    return mv

def calc_temp(value):
    mv = float(value * 1200) / 1023
    degc = ((mv - 500.0) / 10.0) - 4
    #degf = (degc * 1.8) + 32.0
    return degc

def calc_humidity(value):
    mv = float(value * 1200) / 1023
    humidity = (((mv * 108.2 / 33.2) / 5000 - 0.16) / 0.0062)
    return humidity

#####
#Supprime le socket de client,ferme et ensembles à none.
#####
def cleanup(client):
    client_list.remove(client)
    sock = sock_client[client]
    sock_client[sock] = None
    del sock_client[client]

    client.close()
    client = None

def main():
    print "Creating lookup table"
    dest_list = bind_table.node_list.keys() # Obtenir les adresses de XBee

    # Permettre la recherche inversée
    for dest in dest_list:
        port = bind_table.node_list[dest]
        bind_table.node_list[port] = dest

    # Déclarer le socket de TCP
```

```

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# L'adresse peut répéter
sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
sock.bind(("", port))
sock.settimeout(0)
sock.listen(5)

sock_port[sock] = port
sock_port[port] = sock

sock_client[sock] = None
listen_list.append(sock)

# Commencer le socket de XBee
print "Creating xbee socket"
# print "Packet size is: 72 "
# Déclarer le socket de XBee
xb_sock = socket.socket(socket.AF_XBEE, socket.SOCK_DGRAM,
socket.XBS_PROT_TRANSPORT)
xb_sock.bind(("", 0xe8, 0xc105, 0x92)) # zig_sock.bind(("", end_point,
# profile_id, cluster_id))

xb_sock.settimeout(0)

TCP_BUFFERING_TIME = .300 #Time to buffer TCP packets, helps with
zigbee fragmentation and may reduce TCP cost to send data

# Si la connexion de TCP a exception
quit_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
quit_sock.bind(("", 40001))
quit_sock.listen(1)

xb_queue = [] # Une liste pour mettre la commande de TCP à SmartPlug

print "Entering mainloop"
while 1:
    read_list = [xb_sock] + listen_list + client_list + [quit_sock]
    write_list = [xb_sock] + client_list
    rl, wl, el = select.select(read_list, write_list, [], 1.0)

    # XBee présente dans la liste de lire, nous avons les données de XBee
    if xb_sock in rl:
        data, addr = xb_sock.recvfrom(8192)
        print "Received %d bytes from address: " %len(data), addr
        src = addr[0][1:24] # src est l'adresse de XBee
        print "sp addr is: %s " %src
        src_ddo = "[%s]!" %src # src_ddo est un format pour

```



```

D4 = ddo_get_param(src_ddo, 'D4')      # « ddo_get_param() »
if D4 != 0:                             # Obtenir les données de D4
    D4 = int(ord(D4))
    print "D4 is %s"%D4

if len(data) == 12:                     # Les données comprennent les
trois                                  # valeurs de capteur

# Si D4 != 0, le XBee est un SmartPlug, on peut débiller le paquet de
# données et calculer les valeurs de lumière, de température et de courant.
if D4 != 0:
    fixed_byte, digital_bits, analog_set, dont_know_bit1, dont_know_bit2,
    lig, tmp, current = struct.unpack(">bhbbbhhh", data[:12])

    print fixed_byte, digital_bits, analog_set, dont_know_bit1,
    dont_know_bit2, lig, tmp, current

    light = calc_light(lig)
    temp = calc_temp(tmp)
    current = calc_current(current)

    print "lig of smart plug is %4.2f"%light
    print "temp is %4.2f" %temp
    print "current is %4.2f" %current

    list_info = [D4, light, temp, current] # Une liste pour enregistrer les
                                           # valeurs de smartplug, afin de
                                           # la envoyer à TCP

    list_info = list_info[-4:]

else:                                   # D4 = 0, XBee est un capteur,
                                           # il a les valeurs de lumière, de
                                           # température, et de humidité
    fixed_byte, digital_bits, analog_set, dont_know_bit1, dont_know_bit2,
    lig, tmp, humidity = struct.unpack(">bhbbbhhh", data[:12])
    print fixed_byte, digital_bits, analog_set, dont_know_bit1,
    dont_know_bit2
    light = calc_light(lig)
    temp = calc_temp(tmp)
    humidity = calc_humidity(humidity)

    print "lig of smart plug is %4.2f"%light
    print "temp is %4.2f" %temp
    print "humidity is %4.2f" %humidity

```

```

list_info = [light, temp, humidity]    # Enregistrer les valeurs de
list_info = list_info[-3:]             # capteur dans la liste

xbee_dic[src]=list_info                # Enregistrer la liste et
                                        # l'adresse dans le dictionnaire

print (xbee_dic)
sp_info = repr(xbee_dic)               # Transformer le dictionnaire à
                                        # string, parce que le socket ne
                                        # peut pas envoyer un
                                        # dictionnaire

xbee_addr = xbee_dic.keys()
print xbee_addr

# XBee présente dans la liste d'écrire, il y a une commande à SmartPlug
if xb_sock in wl and len(xb_queue) != 0:
    print xb_queue[0]
    data, addr = xb_queue[0]           # 'data' est le données envoyer par TCP

    send_size = (len(data) < 72) and len(data) or 72
    print "Writing %d bytes to dest: %s" %(send_size, str(dest))
    print data

#####
# D1 est la première smartplug dans la liste xbee_addr, D2 est la deuxième.
# 4 représente le command pour fermer la sortie de courant, 5 représente le
# command pour ouvrir le sortie de courant.
#####
try:
    if data == 'D1:4':
        src_ddo = "[%s]!"%xbee_addr[1]
        ddo_set_param(src_ddo, 'D4', 4)    # Configurer D4 à 4 de samrtplug1
    if data == 'D1:5':
        src_ddo = "[%s]!"%xbee_addr[1]
        ddo_set_param(src_ddo, 'D4', 5)    # Configurer D4 à 5 de samrtplug1

    if data == 'D2:4':
        src_ddo = "[%s]!"%xbee_addr[2]
        ddo_set_param(src_ddo, 'D4', 4)    # Configurer D4 à 4 de samrtplug2
    if data == 'D2:5':
        src_ddo = "[%s]!"%xbee_addr[2]
        ddo_set_param(src_ddo, 'D4', 5)    # Configurer D4 à 5 de samrtplug2

    xb_queue.pop(0)                      # Vider la liste

```

```

except Exception, e:
    print "Exception occured sending to address: %s" %(str(addr))
    xb_queue.pop(0)
    port = bind_table.node_list[addr]
    sock = sock_port[port]

for sock in listen_list:
    # sock dans la liste de lire, indique que il y a des client connectés
    if sock in rl:
        client, addr = sock.accept()
        print "Received TCP connection from address: ", addr

        # Permettre la recherche inversée
        sock_client[sock] = client
        sock_client[client] = sock

        sock_queue[sock] = []
        client_list.append(client)

for client in client_list:
    if client in rl:
        print "Reading from client"
        try:
            data = client.recv(8192)      # Recevoir la donnée de TCP
            print "Received %d bytes from TCP connection" %len(data)
        except Exception, e:
            print e
            cleanup(client)
            continue

        if len(data) == 0:                # Si il n'y a pas de donnée de TCP,
            cleanup(client)              # supprime l'adresse
            continue

        sock = sock_client[client]
        port = sock_port[sock]
        xb_queue.append((data, bind_table.node_list[port]))

for client in client_list:
    if client in wl:
        client.send(sp_info)             # Envoyer le string de dictionnaire qui
                                         # contient les adresses et les valeurs de Xbee

if quit_sock in rl:
    raise KeyboardInterrupt("Stopping script, quit sock activated")

```

```
if __name__ == '__main__':
    main()
```

iii. HTML et le serveur de Web

```
#####
# Le code de ce programme contient deux parties, une d'elles vise à
# développer l'interface de Web, l'autre partie est pour créer un
# serveur de web.
#####

from wsgiref.simple_server import make_server
from cgi import parse_qs, escape
import socket
import struct
import json

#####
#HTML
#####
html = """
<html>
<head>

<!--Tous les contenus sont mis dans un 'container' -->

<style type="text/css">
div#container{width:700px;
    height:800px;
    border-style:groove;
    border-width:10px;
    padding:20px;
    margin:100px auto;
    font-size:17px}
div#header {background-color:#ffffcc;}

<!-- Séparer le conteneur à quatre parties, définir leur styles -->
div#content_left_sensor {background-color:#f0f0f0;
    width:300;
    height:182;
    float:left;
    text-align:right;
```

```

        overflow:auto;}
div#content_left_up {background-color:#f0f0f0;
    width:300;
    height:260;
    float:left;
    text-align:right;
    overflow:auto;}
div#content_left_down {background-color:#f0f0f0;
    width:300;
    height:260px;
    float:left;
    text-align:right;}
div#content_right_sensor {background-color:#f0f0f0;
    width:395;
    height:182px;
    overflow:auto;
    float:right;
    }
div#content_right_up {background-color:#f0f0f0;
    width:395;
    height:260px;
    overflow:auto;
    float:right;
    }
div#content_right_down {background-color:#f0f0f0;
    width:395;
    height:260px;
    overflow:auto;
    float:right;}
<!--
.green{background-color:green;}
.red{background-color:red;}
-->
</style>
</head>

<body>
<div id="container">

    <div id="header">
        <h1 align="center">Smart Home Control</h1>
    </div>
    <div id="content_left_sensor">
        <br>
        Sensor addr is:<br><br>
        Temperature (C):<br><br>

```

```

    Light (Lux):<br><br>
    Humidity (RH):<br><br>
</div>
<div id="content_right_sensor">
    <br>
    <form id="sensor_info" method="get" action="">
        <input type="text" name="sensor_addr" value="%s" /><br><br>
        <input type="text" name="sensor_temp" value="%s" /><br><br>
        <input type="text" name="sensor_light" value="%s" /><br><br>
        <input type="text" name="sensor_current" value="%s" />
    </form>
</div>
<hr style="border:1px dashed #FFFFFF;border-bottom:0;border-right:0;
        border-left:0;width:600px;">
<div id="content_left_up">
    <br>
    Smart plug_1 addr is:<br><br>
    Temperature (C):<br><br>
    Light (Lux):<br><br>
    Current (A):<br><br>
    Control(On/Off):<br>
    <br><br>
</div>
<div id="content_right_up">
    <br>
    <form id="sp1_info" method="get" action="">
        <input type="text" name="sp1_addr" value="%s" /><br><br>
        <input type="text" name="sp1_temp" value="%s" /><br><br>
        <input type="text" name="sp1_light" value="%s" /><br><br>
        <input type="text" name="sp1_current" value="%s" />
    </form>
    <form>
        <input type="text" id="sp1_state" value="%s" style="display:none"/>
        <input type="text" id="spgreen_1" class="green" style="height:15px;
width:20px; background-color:green"/>
        <input type="text" id="spred_1" class="red" style="height:15px;
width:20px; background-color:red" />
    <script>
        var x=document.getElementById("sp1_state").value
        if (x!=null)
        {
            if (x == 4)
                {document.getElementById("spgreen_1").style.display="none";}
            else if(x == 5)
                {document.getElementById("spred_1").style.display="none";}
        }
    </script>

```

```

else
{
    document.getElementById("spgreen_1").style.display="none";
}
</script>

<input type="radio" name= "control_1" value="5" /> On
<input type="radio" name= "control_1" value="4" /> Off
<br><br>
<input type="submit" value="Submit" /><br>
</form>
</div>
<hr style="border:1px dashed #FFFFFF;border-bottom:0;border-right:0;
        border-left:0;width:600px;">
<div id="content_left_down">
<br>
Smart plug_2 addr is:<br><br>
Temperature (C):<br><br>
Light (Lux):<br><br>
Current (A):<br><br>
Control(On/Off):<br><br>
</div>

<div id="content_right_down">
<br>
<form id="sp2_info" method="get" action="">
    <input type="text" name="sp2_addr" value="%s" /><br><br>
    <input type="text" name="sp2_temp" value="%s" /><br><br>
    <input type="text" name="sp2_light" value="%s" /><br><br>
    <input type="text" name="sp2_current" value="%s" />
</form>
<form>
    <input type="text" id="sp2_state" value="%s" style="display:none"/>
    <input type="text" id="spgreen_2" class="green" style="height:15px;
width:20px; background-color:green"/>
    <input type="text" id="spred_2" class="red" style="height:15px;
width:20px; background-color:red" />

<script>
var x=document.getElementById("sp2_state").value
if (x!=null)
{
    if (x == 4)
        {document.getElementById("spgreen_2").style.display="none";}
    else if(x == 5)
        {document.getElementById("spred_2").style.display="none";}
}

```

```

    }
    else
    {
        document.getElementById("spgreen_2").style.display="none";
    }
</script>

<input type="radio" name= "control_2" value="5" /> On
<input type="radio" name= "control_2" value="4" /> Off
<br><br>
<input type="submit" value="Submit" />
</form>
</div>

</div>
</body>
</html>
"""

#####
# L'application pour créer un serveur et traiter les données
#####
def application(envIRON, start_response):
    print "QUERY_STRING: %s" %envIRON['QUERY_STRING']
    print "REQUEST_METHOD: %s" %envIRON['REQUEST_METHOD']

    # Retourner un dictionnaire qui comprend les listes comme valeur
    d = parse_qs(envIRON['QUERY_STRING'])

    # Déclarer le socket
    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect(('132.209.9.41', 4000))    # Connecter à la passerelle
        print "s connected"
    except:
        print "socket wrong"

    try:
        data = eval(s.recv(2048))
        addr = data.keys()
        print addr

        #####
        # En cas du capteur n'a plus de charge, le serveur va recevoir
        # les données seulement de smartplug, la page web va montrer 'empty'
        # dans la zone de capteur
        #####

```



```

if len(addr) == 3:
    addr_sensor = addr[0]
    addr_1 = addr[1]
    addr_2 = addr[2]
    print "sensor addr is: %s" %addr_sensor
    print "smartplug1 addr is: %s" %addr_1
    print "smartplug2 addr is: %s" %addr_2

    light_sensor = data.get(addr_sensor)[0]
    temp_sensor = data.get(addr_sensor)[1]
    humidity_sensor = data.get(addr_sensor)[2]
    print light_sensor,temp_sensor, humidity_sensor

    D4_1 = data.get(addr_1)[0]
    light_1 = data.get(addr_1)[1]
    temp_1 = data.get(addr_1)[2]
    current_1 = data.get(addr_1)[3]
    print D4_1, light_1, temp_1, current_1

    D4_2 = data.get(addr_2)[0]
    light_2 = data.get(addr_2)[1]
    temp_2 = data.get(addr_2)[2]
    current_2 = data.get(addr_2)[3]
    print D4_2, light_2, temp_2, current_2

    sensor_temp = round(temp_sensor,3)
    sensor_lig = round(light_sensor,3)
    sensor_humidity = round(humidity_sensor,3)
else:
    addr_sensor = None
    sensor_temp = None
    sensor_lig = None
    sensor_humidity = None
    addr_1 = addr[0]
    addr_2 = addr[1]
    print "smartplug1 addr is: %s" %addr_1
    print "smartplug2 addr is: %s" %addr_2

    D4_1 = data.get(addr_1)[0]
    light_1 = data.get(addr_1)[1]
    temp_1 = data.get(addr_1)[2]
    current_1 = data.get(addr_1)[3]
    print D4_1, light_1, temp_1, current_1

    D4_2 = data.get(addr_2)[0]
    light_2 = data.get(addr_2)[1]

```

```

        temp_2 = data.get(addr_2)[2]
        current_2 = data.get(addr_2)[3]
        print D4_2, light_2, temp_2, current_2
    except:
        print "recv wrong"

# Garder deux chiffres après virgule fractionnaire
sp1_temp = round(temp_1,3)
sp1_lig = round(light_1,3)
sp1_current = round(current_1,3)

sp2_temp = round(temp_2, 3)
sp2_lig = round(light_2,3)
sp2_current = round(current_2,3)

# Obtenir la commande de client, et envoyer les données à la passerelle
control_1 = d.get('control_1', [])
print control_1
if control_1 == ['4']:
    s.send('D1:4')
if control_1 == ['5']:
    s.send('D1:5')

control_2 = d.get('control_2', [])
print control_2
if control_2 == ['4']:
    s.send('D2:4')
if control_2 == ['5']:
    s.send('D2:5')

# Enregistrer les données à 'response_body'
response_body = html % (addr_sensor or 'Empty', sensor_temp or 'Empty',
                        sensor_lig or 'Empty', sensor_humidity or 'Empty',
                        addr_1 or 'Empty', sp1_temp or 'Empty', sp1_lig or 'Empty',
                        sp1_current or 'Empty', D4_1 or 'Empty',
                        addr_2 or 'Empty', sp2_temp or 'Empty', sp2_lig or 'Empty',
                        sp2_current or 'Empty', D4_2 or 'Empty')

status = '200 OK'

# Le type est text/html, répondre les contenus à la page
response_headers = [('Content-Type', 'text/html'),
                    ('Content-Length', str(len(response_body)))]
start_response(status, response_headers)
return [response_body]

```

```
#s.close()

# Créer le serveur qui avec l'adresse de l'ordinateur et la porte 8000
httpd = make_server("", 8000, application)
httpd.serve_forever()
```